Nonsymmetric Multilevel RBF Collocation within an Operator Newton Framework for Nonlinear PDEs

Gregory E. Fasshauer

Abstract. We present a multilevel algorithm based on nonsymmetric collocation with compactly supported radial basis functions for the solution of nonlinear partial differential equations. At the operator level this algorithm can be viewed as a Newton method. Details of the algorithm as well as numerical experiments are discussed.

§1. Introduction

Radial basis function (RBF) methods have been studied by various authors over the past two decades. Most of the research (theoretical and applied) has focussed on the use of RBFs in approximation theory (i.e., scattered data approximation). More recently, however, some authors have also considered using RBFs for the numerical solution of partial differential equations (PDEs). An overview of some of these approaches is given in [3].

In this paper we will show how RBF collocation (briefly explained in Section 3) can be applied to the solution of nonlinear PDEs. More specifically, we will begin in Section 2 by reviewing an operator Newton framework discussed in detail in [6], and then couple this approach with compactly supported RBFs (briefly discussed in Section 4). A short motivation of why the kind of multilevel/Newton framework suggested here is the natural environment in which to employ compactly supported RBFs follows in Section 5. Our suggested approach leads to an algorithm whose details we discuss in Section 6, and whose performance we illustrate in Section 7. Some closing remarks are provided in Section 8.

Trends in Approximation Theory
Kirill Kopotun, Tom Lyche, and Mike Neamtu (eds.), pp. 1–10.

ISBN 0-8265-xxxx-x.

§2. An Operator Newton Method

We consider a generic nonlinear (elliptic) PDE of the form $\mathcal{L}u = f$ in $\Omega \subset \mathbb{R}^d$. A general Newton framework was described in [9,6]. The iteration for the approximate solution u_k proceeds as follows:

$$u_k - u_{k-1} = -T_{h_k}(u_{k-1})F(u_{k-1}), \qquad k \geq 1, \tag{1}$$

where u_0 is some initial guess, $F(u) = \mathcal{L}u - f$ computes the residual, and T_{h_k} is a numerical inversion operator (i.e., T_{h_k} approximates $(F')^{-1}$ on a mesh with meshsize h_k). Thus the Newton method is used to find the zero of the operator F. The discussion in [9,6] is quite general and allows for many different kinds of numerical inversion techniques. In this paper we will concentrate on RBF collocation, but, e.g., in [5] we have employed a finite element Galerkin method.

The main result of [9,6] is that for many numerical inversion methods the Newton method will suffer from a loss of derivatives, i.e., not achieve full (quadratic) convergence, and that if an appropriate smoothing is incorporated at each iteration, then almost full (superlinear) convergence can be achieved. In this paper we will observe the former (degraded convergence), but not make a serious attempt to find an appropriate smoothing. Through a simple scaling technique convergence is maintained. The price, however, is greater computational cost.

§3. Nonsymmetric RBF Collocation

The following approach to solving (elliptic) PDEs by collocation goes back to Kansa [10]. We illustrate the method with a linear(ized) boundary value problem of the form

$$\begin{aligned} Lu &= f \qquad \text{in } \Omega \subset \mathbb{R}^d, \\ Bu &= g \qquad \text{on } \partial\Omega. \end{aligned} \tag{2}$$

The idea is to use the *Ansatz*

$$u(x) = \sum_{j=1}^{n} c_j \phi(\|x - x_j\|), \tag{3}$$

where the points $x_j \in \mathbb{R}^d$ are referred to as centers or knots, n_I of which lie in the interior of Ω and n_B on the boundary. If one now collocates at the same set of points, i.e., enforces the conditions (2) at x_i, $i = 1, \ldots, n$, then a system of linear equations $Ac = y$ arises with

$$A = \begin{bmatrix} L\Phi \\ B\Phi \end{bmatrix}, \qquad c = \begin{bmatrix} c_I \\ c_B \end{bmatrix}, \qquad y = \begin{bmatrix} f \\ g \end{bmatrix}. \tag{4}$$

Here $L\Phi$ is an $n_I \times n$ matrix with entries $L\Phi_{ij} = L\phi(\|x - x_j\|)|_{x=x_i}$, $B\Phi$ is an $n_B \times n$ matrix with entries $B\Phi_{ij} = B\phi(\|x - x_j\|)|_{x=x_i}$, and c_I and c_B are those coefficients in (3) corresponding to interior and boundary knots, respectively. The pieces f and g of y are obtained analogously.

In [1] this nonsymmetric collocation approach was contrasted with a symmetric one for which it is known that the matrix A is invertible for (distinct) knots in arbitrary positions provided ϕ is taken from an appropriate class of functions (e.g., strictly positive definite functions). For the nonsymmetric method it is now known that there exist configurations of knots for which A is singular [8]. Since the nonsymmetric approach is easier to implement (especially for nonlinear problems), and since singular configurations seem to be quite rare (cf. [8]), we have elected to base our present discussion on this method. A further motivation is given by the entries in Table 2 in Section 5.

§4. Wendland's Compactly Supported RBFs

Earlier papers on RBF collocation used globally supported RBFs such as multiquadrics, thin plate splines, or Gaussians. These methods, however, lead to dense collocation matrices A, and are thus not viable for large problems without further work such as domain decomposition or fast evaluation techniques. Compactly supported RBFs, on the other hand, lead to sparse matrices and are therefore feasible for large problems. The most popular such class of functions was introduced by Wendland [12], and explicit formulas for C^0, C^2, C^4, and C^6 functions $\phi_{\ell,\kappa}$ were given in [2]:

$$\begin{aligned}
\phi_{\ell,0}(r) &\doteq (1-r)_+^{\ell} \\
\phi_{\ell,1}(r) &\doteq (1-r)_+^{\ell+1}\left[(\ell+1)r+1\right] \\
\phi_{\ell,2}(r) &\doteq (1-r)_+^{\ell+2}\left[(\ell^2+4\ell+3)r^2+(3\ell+6)r+3\right] \\
\phi_{\ell,3}(r) &\doteq (1-r)_+^{\ell+3}\big[(\ell^3+9\ell^2+23\ell+15)r^3 \\
&\qquad +(6\ell^2+36\ell+45)r^2+(15\ell+45)r+15\big]\,.
\end{aligned} \tag{5}$$

Here $\doteq$ stands for equality up to a constant factor, $r = \|x - \cdot\|$, and $\ell = \lfloor \frac{d}{2} \rfloor + \kappa + 1$, where d is the space dimension and κ denotes the smoothness. In particular, the function $\phi_{\ell,\kappa}$ is known to be (strictly) positive definite in $\mathbb{R}^d$ (and any lower-dimensional Euclidean space), and to lie in the smoothness class $C^{2\kappa}$. In our numerical examples below we have used the function

$$\phi_{5,3}(r) = (1-r)_+^8(32r^3+25r^2+8r+1) \tag{6}$$

which is C^6 and positive definite in $\mathbb{R}^3$. The choice of this specific function is motivated by the fact that symmetric collocation (used in Tables 1 and 2 below) requires more smoothness than the nonsymmetric method, and we want to make a uniform choice of ϕ throughout the paper.

The functions listed in (5) are supported on the interval $[-1,1]$, or after composition with the Euclidean norm, on the unit ball in $\mathbb{R}^d$. The multilevel algorithm will make use of scaled versions of these functions, i.e., $\phi_{\ell,\kappa,\alpha}(r) = \phi_{\ell,\kappa}(r/\alpha)$, which are supported on $[-\alpha,\alpha]$ or a ball with radius α in $\mathbb{R}^d$.

§5. Why Multilevel?

In [7] a multilevel scattered data interpolation algorithm based on compactly supported RBFs was suggested. Its connection to Newton iteration was explained in [6]. We now give a brief argument why this kind of multilevel algorithm is "natural" when using compactly supported RBFs.

Let us first consider the solution of a linear two-point boundary value problem $-u''(x) + \pi^2 u(x) = 2\pi^2 \sin \pi x$, $x \in (0,1)$, $u(0) = u(1) = 0$, by symmetric RBF collocation. Due to space limitations we do not give any details for the symmetric collocation method here (see e.g. [1]). The entries in the tables suffice to make our point (a similar argument for scattered data interpolation was given in [11]). First we investigate what happens if we use (6) in a series of approximations on increasingly finer meshes. The size of the support of the basis functions (starting with $\alpha = 2$) is scaled proportional to the data density, i.e., since h is decreased to $h/2$ from one experiment to the next, so is the scale α for the support of ϕ. This causes the bandwidth B of the collocation matrix A to remain constant throughout the series of experiments ($B = 33$). From the left part of Table 1 we can see that this method of approximation is extremely efficient (=fast) for large amounts of data, but very inaccurate. On the other hand, if we keep the size of the support fixed throughout the series of approximations, i.e., the matrices are always full matrices, then the approximations on finer data sets are extremely accurate, but very expensive to compute. In fact, the asterisk in the last row of Table 1 indicates that the conjugate gradient method used for the solution of the linear system used up its maximum allowed number of iterations (we used LASPACK by Tomas Skalicky with a maximum of 10000 allowed iterations) before it could reach the required accuracy.

Mesh	Fixed bandwidth			Full matrices	
	ℓ_2-error	time	$\% \neq 0$	ℓ_2-error	time
17	$5.408189\,10^{-6}$	0	100	$5.408189\,10^{-6}$	0
33	$1.869438\,10^{-4}$	0	78	$3.409848\,10^{-8}$	0
65	$2.410870\,10^{-2}$	0	46.4	$1.740636\,10^{-10}$	0
129	$3.320252\,10^{-1}$	0	25.3	$7.804604\,10^{-13}$	1
257	$4.852515\,10^{-1}$	1	13.2	$3.260658\,10^{-15}$	11
513	$4.986328\,10^{-1}$	1	6.7	$1.672224\,10^{-16}$	129
1025	$4.997009\,10^{-1}$	3	3.4	$6.547368\,10^{-15*}$	1196

Tab. 1. Collocation on single grids. Fixed bandwidth and full matrices.

The entries in Table 1 should be compared to those in Table 2. There we have used a multilevel algorithm (more details in Section 6) with symmetric as well as nonsymmetric RBF collocation. The entries in the "rate" column correspond to the ρ in convergence rates of the form $O(h^\rho)$, i.e., if the meshsize is decreased by a factor of 2 the error decreases by a factor of 2^ρ. Clearly, the multilevel approach leads to good results for the errors as well as for the execution times. It is apparent from the table that the nonsymmetric collocation method performs much better than the symmetric one. The deteriorating convergence property of the symmetric collocation approach can be attributed to the loss of derivatives mentioned in Section 2, and represents the additional motivation for us to use nonsymmetric collocation as mentioned at the end of Section 3. We note, however, that we have encountered similar behavior with the nonsymmetric approach – only at later stages.

	symmetric			nonsymmetric			
Mesh	ℓ_2-error	rate	time	ℓ_2-error	rate	time	$\% \neq 0$
17	$5.408189\,10^{-6}$		0	$1.113709\,10^{-6}$		0	100
33	$1.160280\,10^{-7}$	5.54	0	$2.421506\,10^{-8}$	5.52	0	77
65	$1.548268\,10^{-8}$	2.91	1	$5.862669\,10^{-10}$	5.37	0	45.4
129	$1.118532\,10^{-8}$	0.47	1	$1.458817\,10^{-11}$	5.33	0	24.6
257	$1.093402\,10^{-8}$	0.03	2	$3.659998\,10^{-13}$	5.32	1	12.8
513	$1.092675\,10^{-8}$	0.00	2	$9.262179\,10^{-15}$	5.30	1	6.5
1025	$1.093092\,10^{-8}$	0.00	4	$2.414450\,10^{-16}$	5.26	3	3.3

Tab. 2. Multilevel collocation with fixed bandwidth.

§6. The Multilevel Newton Algorithm for $\mathcal{L}u = f$

Our multilevel algorithm follows the general Newton iteration (1).

Algorithm 1.

1) *Create nested knot sets* $\mathcal{X}_1 \subset \cdots \subset \mathcal{X}_K \subset \Omega$, *and pick an initial guess* u_0.
2) *For* $k = 1, 2, \ldots, K$
   a) *Solve the linearized problem*

$$L_{u_{k-1}} v = f - \mathcal{L}u_{k-1} \quad \text{on } \mathcal{X}_k.$$

   b) *Perform the Newton update of the* k*-th iterate*

$$u_k = u_{k-1} + v.$$

Newton iteration (1) coupled with a smoothing of the Newton update was referred to in [9,6,4] as Nash iteration. In [4] various interpretations of Nash iteration were discussed and implemented for linear problems. Algorithm 1 is a version of the simplest interpretation of Nash iteration (Algorithm 2.1 in [4]) – but without the smoothing.

For the numerical experiments presented in the next section we used the nonlinear PDE

$$\begin{aligned} -\varepsilon^2 \nabla^2 u - u + u^3 &= f \qquad \text{in } \Omega = (0,1) \times (0,1), \\ u &= 0 \qquad \text{on } \partial\Omega. \end{aligned} \tag{7}$$

This equation was also used in [5] where the performance of Algorithm 1 was discussed when using a piecewise linear finite element Galerkin method for the solution of the linearized problem at each step. The right-hand side f was chosen so that (7) has an analytic solution of the form

$$u(\xi,\eta) = \psi(\xi)\psi(\eta) \tag{8}$$

with

$$\psi(t) = 1 + e^{-1/\varepsilon} - e^{-t/\varepsilon} - e^{(t-1)/\varepsilon}. \tag{9}$$

Here (ξ,η) denote the Cartesian coordinates of $x \in \mathbb{R}^2$, and the parameter ε determines the size of the boundary layers near the edges of the domain Ω. A more detailed discussion of this problem is given in [5].

For this model problem the linearized equation to be solved in step 2a of Algorithm 1 is of the form

$$-\varepsilon^2\nabla^2 v + (3u_{k-1}^2 - 1)v = f + \varepsilon^2\nabla^2 u_{k-1} + u_{k-1} - u_{k-1}^3.$$

In our implementation of Algorithm 1 we keep track of the approximate solution u on an extra fine grid $\mathcal{X}_{K+1}$ on which we also compute the errors listed in the tables below. At each step k we solve the following linear system arising from the nonsymmetric collocation procedure (2)–(4):

$$\sum_{j=1}^{n} c_j \left[-\varepsilon^2[\nabla^2\phi]_{ij} + (3u_i^2 - 1)\phi_{ij}\right] = f + \varepsilon^2[\widetilde{\nabla}^2 u]_i + u_i - u_i^3, \quad i = 1,\ldots,n_I,$$
$$\sum_{j=1}^{n} c_j\phi_{ij} = 0, \quad i = 1,\ldots,n_B, \tag{10}$$

where, for transparency, we have omitted the index k on the quantities n, n_I, n_B, and c_j. The other abbreviations used in (10) are $\phi_{ij} = \phi_{5,3,\alpha_k}(\|x_i - x_j\|)$, $[\nabla^2\phi]_{ij} = \left(\frac{\partial^2}{\partial\xi^2} + \frac{\partial^2}{\partial\eta^2}\right)\phi_{5,3,\alpha_k}(\|x - x_j\|)|_{x=x_i}$, and $u_i = u_{k-1}(x_i)$, and $[\widetilde{\nabla}^2 u]_i$ is used to denote the usual discretized Laplacian of u_{k-1} evaluated at x_i. The points x_i and x_j are taken from the knot set $\mathcal{X}_k$. The Newton update v is then given by

$$v(x) = \sum_{j=1}^{n^{(k)}} c_j^{(k)}\phi_{5,3,\alpha_k}(\|x - x_j^{(k)}\|).$$

§7. Numerical Examples

We now present some sample results obtained using the method described in the previous section. As knot sets $\mathcal{X}_k$, $k = 1,\ldots,K$, we have used uniformly spaced points in the unit square $[0,1]\times[0,1]$. The number of points used in each iteration is listed in the N-column of Tables 3–6. As already mentioned above we used the RBF $\phi_{5,3}$ listed in (6). For the solution of the nonsymmetric (but sparse) linear systems we used the conjugate gradient method applied to the normal equations coupled with incomplete LU-preconditioning. The other entries in the tables are the ℓ_2 errors computed on the grid $\mathcal{X}_{K+1}$ of 66049 uniformly distributed points in the unit square, the "bandwidth" B of the matrix A (roughly the number of knots inside the support of the basis functions), the convergence rate (as explained in Section 5), the execution time in seconds, and the percentage of nonzero elements in A.

For the first experiment (summarized in Table 3) we kept the bandwidth fixed. This implies that since the meshsize h of the knot sets $\mathcal{X}_k$ decreases by a factor of 2 from one iteration to the next, so does the scale α of the support of the basis functions ϕ_α. We can see that the matrices become increasingly sparser and the procedure is relatively fast. However, the same loss of derivatives as mentioned in Section 5 is experienced.

N	ℓ_2-error	rate	B	time	% nonzero
25	$1.179217\,10^{-1}$		81	0	100
81	$6.468158\,10^{-2}$	0.87	81	0	56.7
289	$2.568938\,10^{-2}$	1.33	81	0	21.2
1089	$9.967680\,10^{-3}$	1.37	81	2	6.47
4225	$4.429941\,10^{-3}$	1.17	81	10	1.79
16641	$2.722039\,10^{-3}$	0.70	81	49	0.47

Tab. 3. Nonsymmetric multilevel collocation with fixed bandwidth.

In Tables 4 and 5 we have tried to compensate for this effect by slowly increasing B. The results are as expected. But the $O(h^2)$ convergence in Table 4 comes at a high computational cost. Some fine-tuning of support size and computational cost are required (as attempted for the results given in Table 5).

N	ℓ_2-error	rate	B	time	% nonzero
25	$1.179217\,10^{-1}$		81	0	100
81	$5.378768\,10^{-2}$	1.13	121	1	72.6
289	$1.634885\,10^{-2}$	1.72	169	2	38.4
1089	$4.550885\,10^{-3}$	1.84	169	13	12.6
4225	$1.172617\,10^{-3}$	1.96	169	82	3.61
16641	$2.753521\,10^{-4}$	2.09	169	475	0.97

Tab. 4. Nonsymmetric multilevel collocation with increasing bandwidth.

Finally, the entries in Table 6 were obtained with a similar strategy as those in Tables 4 and 5 starting, however, on a very coarse grid of only the four corner points of the unit square. The results are very similar to those listed in Table 4 with the errors roughly three times larger on corresponding knot sets.

N	ℓ_2-error	rate	B	time	% nonzero
25	$1.369166\,10^{-1}$		49	0	84.6
81	$7.416272\,10^{-2}$	0.88	49	0	39.6
289	$3.033178\,10^{-2}$	1.29	49	0	13.7
1089	$1.128463\,10^{-2}$	1.43	49	1	4.04
4225	$4.064507\,10^{-3}$	1.48	81	12	1.79
16641	$1.406662\,10^{-3}$	1.53	81	77	0.47

Tab. 5. Nonsymmetric multilevel collocation with increasing bandwidth.

N	ℓ_2-error	rate	B	time	% nonzero
4	$2.187407\,10^{-1}$		9	0	100
9	$2.217119\,10^{-1}$	-0.02	9	0	60.5
25	$1.979660\,10^{-1}$	0.16	9	0	27.0
81	$1.522843\,10^{-1}$	0.38	25	0	23.2
289	$6.709959\,10^{-2}$	1.18	25	0	7.47
1089	$1.752656\,10^{-2}$	1.94	49	2	4.04
4225	$4.436804\,10^{-3}$	1.98	81	13	1.79
16641	$9.080982\,10^{-4}$	2.29	169	412	0.97

Tab. 6. Nonsymmetric multilevel collocation with increasing bandwidth.

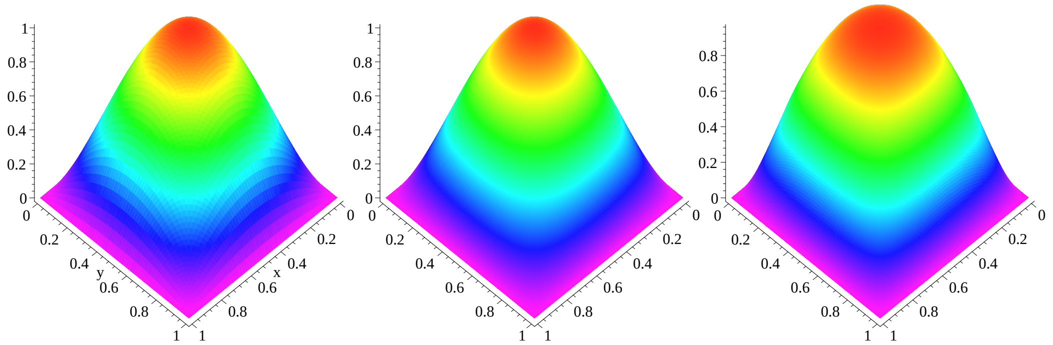

Fig. 1. Initial guess, iterate 1 and iterate 4.

Figures 1 and 2 show perspective plots of some of the iterates along with the initial guess $u_0(\xi,\eta) = 16\xi(1-\xi)\eta(1-\eta)$ and the exact solution (8) and (9).

In Figures 3 and 4 we have plotted the Newton updates at four different steps. It is apparent that the most inaccurate region is along the boundary of the domain. This is quite common when using RBF methods. We hope that the addition of an appropriate smoothing step will be able to alleviate this problem (similar to our observations in [6]). Another possibility, of course, is to add more knots near the boundary, or to use basis functions with varying support radii within one particular level.

§8. Closing Remarks

To our knowledge this presents the first discussion of the use of multilevel RBF

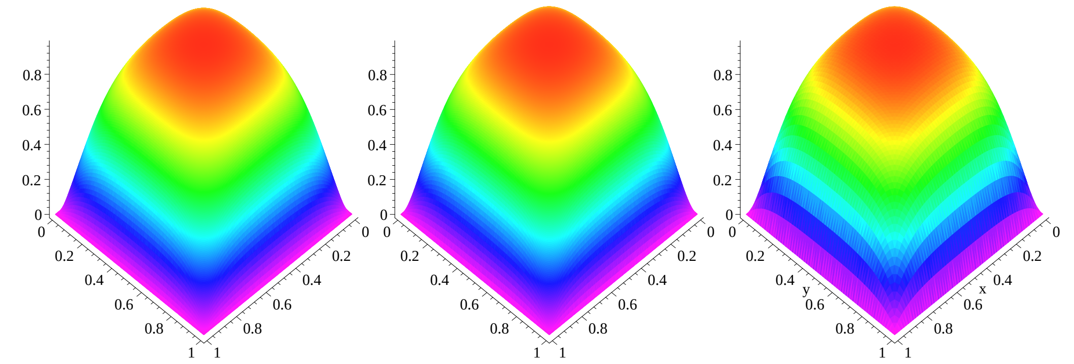

Fig. 2. Iterate 6, iterate 8, and exact solution.

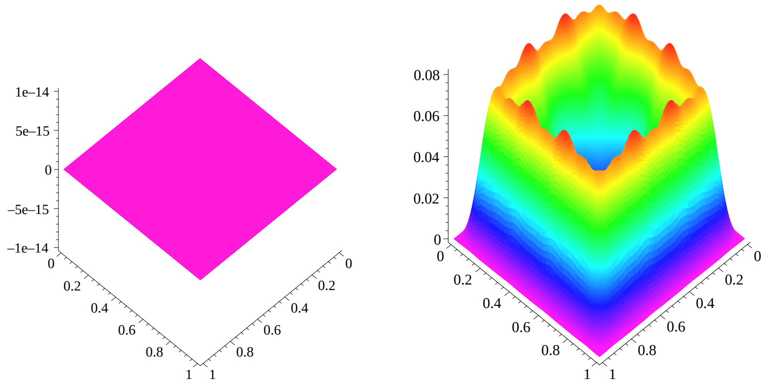

Fig. 3. Newton updates in iteration 1 and 4.

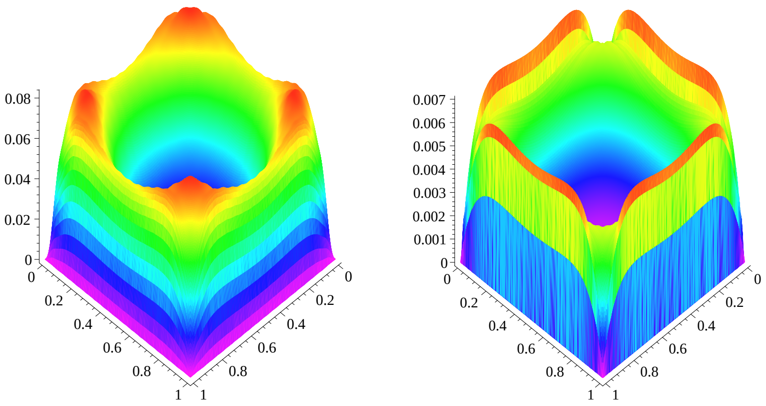

Fig. 4. Newton updates in iteration 6 and 8.

collocation for the solution of nonlinear PDEs. RBF collocation is a fairly simple method to implement (also in higher space dimensions). However, a drawback of our method is the degradation of the convergence properties in later stages of the iteration. This effect was not noticed when using piecewise linear finite elements in [5]. It is our hope that an appropriate smoothing can be found which will serve as a remedy.

References

1. Fasshauer, G. E., Solving partial differential equations by collocation with radial basis functions, in *Surface Fitting and Multiresolution Methods*, A.

LeMéhauté, C. Rabut, and L. L. Schumaker (eds), Vanderbilt University Press, Nashville TN, 1997, 131–138.

2. Fasshauer, G. E., On smoothing for multilevel approximation with radial basis functions, in *Approximation Theory IX, Vol. 2: Computational Aspects,* Charles K. Chui and Larry L. Schumaker (eds), Vanderbilt University Press, Nashville TN, 1998, 55–62.
3. Fasshauer, G. E., Solving differential equations with radial basis functions: Multilevel methods and smoothing, Advances in Comp. Math. **11** (1999), 139–159.
4. Fasshauer, G. E., E. C. Gartland, and J. W. Jerome, Algorithms Defined by Nash iteration: Some implementations via multilevel collocation and smoothing, J. Comput. Applied Math. **119** (2000), 161–183.
5. Fasshauer, G. E., E. C. Gartland, and J. W. Jerome, Newton iteration for partial differential equations and the approximation of the identity, Numer. Algorithms, to appear.
6. Fasshauer, G. E. and J. W. Jerome, Multistep approximation algorithms: Improved convergence rates through postconditioning with smoothing kernels, Advances in Comp. Math. **10** (1999), 1–27.
7. Floater, M. S. and A. Iske, Multistep scattered data interpolation using compactly supported radial basis functions, J. Comput. Applied Math. **73** (1996), 65–78.
8. Hon, Y. C. and R. Schaback, On nonsymmetric collocation by radial basis functions, preprint, 1999.
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. Schaback, R., On the efficiency of interpolation by radial basis functions, in *Surface Fitting and Multiresolution Methods,* A. LeMéhauté, C. Rabut, and L. L. Schumaker (eds), Vanderbilt University Press, Nashville TN, 1997, 309–318.
12. Wendland, H., Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Advances in Comp. Math. **4** (1995), 389–396.

Greg Fasshauer
Department of Applied Mathematics
Illinois Institute of Technology
Chicago, IL 60616
`fass@amadeus.math.iit.edu, http://amadeus.math.iit.edu/~fass`