On Smoothing for Multilevel Approximation with Radial Basis Functions

Gregory E. Fasshauer

Abstract. In a recent paper with Jerome we have suggested the use of a smoothing operation at each step of the basic multilevel approximation algorithm to improve the convergence rate of the algorithm. In our original paper the smoothing was defined via convolution, and its actual implementation was done via numerical quadrature. In this paper we suggest a different approach to smoothing, namely the use of a precomputed hierarchy of smooth functions. This essentially reduces the cost of the smoothing to zero.

§1. Background and Motivation

Multilevel approximation with radial basis functions (RBFs) was first suggested in [4], and since then also investigated theoretically in [2,6]. The basic idea is to work with locally supported basis functions at different levels of resolution. This ensures stability and accuracy, something which is difficult to achieve with globally supported RBFs (cf. the well-known *trade-off principle* [7]). One starts with a coarse-level approximation, and then approximates residuals at increasingly finer levels. These residuals are subsequently added to the coarse-level fit resulting in an improved approximation.

In this paper we will illustrate our ideas with an application of the multilevel approach to the solution of a linear differential equation of the form $Lu = f$ with homogeneous boundary conditions. We point out, however, that the same principle can also be applied in the general framework used in [2]. If we use RBFs as approximating functions and attempt to solve the differential equation via collocation, then the approximant $\tilde{u}$ of u should be constructed via the following *Ansatz*

$$\tilde{u}(x) = \sum_{j=1}^{n_B} c_j \Phi_j(x) + \sum_{j=1}^{n_I} d_j L^{(2)} \Phi_j(x). \tag{1}$$

Approximation Theory IX
Charles K. Chui and Larry L. Schumaker (eds.), pp. 1–8.

ISBN 1-xxxxx-xxx-x.

Here $\Phi_j(x) = \varphi(\|x - x_j\|)$ is a typical basis function, and the superscript on $L^{(2)}$ indicates that the differential operator L treats the function Φ as a function of the second variable, *i.e.*, we are differentiating with respect to the knot locations. The symbols n_B and n_I denote the number of boundary and interior collocation points, respectively. For a more detailed discussion of collocation with RBFs and its connection to Hermite interpolation (in the framework of globally supported RBFs on one single computational grid) we refer the reader to [1]. The main reason for the use of expansion (1) is that the resulting collocation matrix will be symmetric and positive definite, and thus invertible. Error bounds for (single-grid) collocation with globally and locally supported RBFs can be found in [5].

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

1. Generate a nested sequence of computational grids $\mathcal{X}_0 \subset \mathcal{X}_1 \subset \cdots \subset \mathcal{X}_K \subset \Omega$.
2. Compute a coarse-level approximation of the form

$$\tilde{u}_0(x) = \sum_{j=1}^{n_B^0} c_j^0 \Phi_j^0(x) + \sum_{j=1}^{n_I^0} d_j^0 L^{(2)} \Phi_j^0(x),$$

   where $n_B^0 + n_I^0$ is the number of collocation points in X_0, and the coefficients c_j^0 and d_j^0 are determined by solving $L\tilde{u}_0 = f$ on the grid X_0 via collocation.
3. For i from 1 to K do
   a. Compute the residual $r_i = f - L\tilde{u}_{i-1}$ on the grid X_i.
   b. Approximate the residual r_i with a function of the form

$$g_i(x) = \sum_{j=1}^{n_B^i} c_j^i \Phi_j^i(x) + \sum_{j=1}^{n_I^i} d_j^i L^{(2)} \Phi_j^i(x),$$

   where the coefficients are found by solving $Lg_i = r_i$ on the grid X_i (which consists of $n_B^i + n_I^i$ collocation points).
   c. Compute the level-i approximation as

$$\tilde{u}_i(x) = \tilde{u}_0(x) + \sum_{k=1}^{i} g_k(x).$$

In [2] the multilevel approximation algorithm was interpreted as a *Newton method*, and it was shown that a smoothing of the residual at each level can improve the rate of convergence. Specifically, this means that step 3.c. above has to be modified as follows:

c'. Compute a smoothing of the approximation to the residual at level i as

$$s_i(x) = S_i g_i(x),$$

and then compute the level-i approximation as

$$\tilde{u}_i(x) = \tilde{u}_0(x) + \sum_{k=1}^{i} s_k(x).$$

In [2] the smoothing was defined as a convolution with a suitable kernel, *i.e.*, $S_i s = \phi_i * s$. The actual implementation of the smoothing used for the numerical examples in [2] was done via a quadrature approximation of the convolution integral. Besides the amount of work involved in computing this integral, this approach has another considerable drawback: after the quadrature the smoothed residual s_i is known only on a discrete set of points (the grid X_i), and therefore one needs to compute a representation of s_i which can be differentiated. This is necessary since the computation of the residual for the following iteration requires an application of L (see 3.a.). This procedure is inefficient and introduces additional errors.

We are therefore motivated to search for a more efficient implementation of the smoothing. The linearity of L along with the fact that $L^{(2)}$ acts on the second variable, the linearity of convolution, and the expansion of the approximation to the residual, s_i, in the form (1) imply

$$\begin{aligned}
S_i s_i(x) &= (\phi_i * s_i)(x) \\
&= \int s_i(y)\phi_i(x-y)dy \\
&= \int \left[\sum_{j=1}^{n_B^i} c_j^i \Phi_j^i(y) + \sum_{j=1}^{n_I^i} d_j^i L^{(2)}\Phi_j^i(y)\right]\phi_i(x-y)dy \\
&= \sum_{j=1}^{n_B^i} c_j^i \int \Phi_j^i(y)\phi_i(x-y)dy + \sum_{j=1}^{n_I^i} d_j^i L^{(2)} \int \Phi_j^i(y)\phi_i(x-y)dy \\
&= \sum_{j=1}^{n_B^i} c_j^i (\Phi_j^i * \phi_i)(x) + \sum_{j=1}^{n_I^i} d_j^i L^{(2)}(\Phi_j^i * \phi_i)(x) \\
&= \sum_{j=1}^{n_B^i} c_j^i \widetilde{\Phi}_j^i(x) + \sum_{j=1}^{n_I^i} d_j^i L^{(2)}\widetilde{\Phi}_j^i(x).
\end{aligned}$$

This means that instead of actually performing the convolution to obtain the smoothed residual s_i at every level of the iteration, we can *a priori* determine a family of smoother basis functions $\{\widetilde{\Phi}_j^i\}$ and then simply evaluate the expansion of g_i in step 3.b. by replacing the basis functions Φ_j^i with the smoother ones $\widetilde{\Phi}_j^i$. The coefficients *remain the same*, and therefore – except for the one-time investment in the construction of the suitable set of smoother basis functions $\{\widetilde{\Phi}_j^i\}$ – the smoothing can be implemented at virtually no additional cost.

§2. Supporting Numerical Evidence

A family $\{\psi_{\ell,k}\}$ of compactly supported RBFs was introduced in [8]. The 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 $2k$. For example, for $d_0 = 3$ we can compute

$$\begin{aligned}
\psi_{4,2}(r) &\doteq (1-r)_+^6(35r^2+18r+3),\\
\psi_{5,3}(r) &\doteq (1-r)_+^8(32r^3+25r^2+8r+1),\\
\psi_{6,4}(r) &\doteq (1-r)_+^{10}(429r^4+450r^3+210r^2+50r+5),\\
\psi_{7,5}(r) &\doteq (1-r)_+^{12}(2048r^r+2697r^4+1644r^3+566r^2+108r+9),\\
\psi_{8,6}(r) &\doteq (1-r)_+^{14}(46189r^6+73206r^5+54915r^4+24500r^3+6755r^2\\
&\qquad +1078r+77).
\end{aligned} \tag{2}$$

Here $r = \|x\|$, and the symbol $\doteq$ is used to express equality up to a positive multiplicative constant. The functions $\psi_{\ell,k}$ are polynomials in r on $[0,1]$ and are in $C^4(\mathbb{R})$, $C^6(\mathbb{R})$, $C^8(\mathbb{R})$, $C^{10}(\mathbb{R})$, and $C^{12}(\mathbb{R})$, respectively. In view of our observations at the end of the previous section, we are motivated to investigate their use as the smoother basis functions $\widetilde{\Phi}_j^i$ used to evaluate the residuals s_i in the multilevel algorithm.

For our numerical illustrations we choose the following very simple one-dimensional boundary value problem:

$$\begin{aligned}
-u''(x) + \pi^2 u(x) &= 2\pi^2 \sin \pi x, \qquad x \in (0,1),\\
u(0) = u(1) &= 0,
\end{aligned} \tag{3}$$

which has solution $u(x) = \sin \pi x$.

We now compare collocation based on the C^4-function $\psi_{4,2}$. In the left half of Table 1 we do not apply the smoothing, in the right half we do, *i.e.*, we use the smoother functions listed above to evaluate the residuals. The nested sequence of computational grids is generated by taking $\mathcal{X}_i$ to consist of $2^{i+2}+1$ equally spaced points in $[0,1]$.

	no smoothing		smoothing	
$\# \mathcal{X}_i$	ℓ_2 error	rate	ℓ_2 error	rate
5	.1373313245		.1373313245	
9	.04127396164	1.734356908	.03828218880	1.842915509
17	.007501933148	2.459897609	.004390590470	3.124186439
33	.0009036081816	3.053493167	.0003109226600	3.819787305
65	.00008189092187	3.463921899	.00002873418154	3.435715902

Table 1. ℓ_2 errors for the approximate solution of (3) using hierarchy (2).

Using a different family of compactly supported functions (to be introduced in the next section) the results for the smoothing are quite remarkable:

	no smoothing		smoothing	
# $\mathcal{X}_i$	ℓ_2 error	rate	ℓ_2 error	rate
5	.3652893821		.3652893820	
9	.2981395993	.2930518106	.2846817499	.3596898097
17	.1851993745	.6869087828	.1351373266	1.074923789
33	.07734116173	1.259770886	.02942014441	2.199549987
65	.01956947148	1.982631642	.001985448971	3.889267140

Table 2. ℓ_2 errors for the approximate solution of (3) using hierarchy (6).

We point out that we have made no attempt to optimize our method. In fact, due to a very poor initial fit the errors in our examples are all quite large. The purpose of these examples is only to illustrate the *effect* of the smoothing, and thus give some numerical justification to the alternative approach to smoothing described in Sect. 1.

§3. A Second Look at Wendland's Functions $\psi_{\ell,k}$

The basic definition of the functions $\psi_{\ell,k}$ in [8] is

$$\psi_{\ell,k}(r) = I^k \psi_{\ell,0}(r), \tag{4}$$

where

$$\psi_{\ell,0}(r) = (1-r)_+^\ell, \qquad \text{and} \qquad If(r) = \int_r^\infty sf(s)ds. \tag{5}$$

We now prove

Theorem 1. *The functions $\psi_{\ell,k}$, $k \geq 1$, can be computed recursively as*

$$\psi_{\ell,k}(r) = \int_r^1 s\psi_{\ell,k-1}(s)ds.$$

Proof: The proof is by induction on k. The induction step is easily obtained using (4), (5), and the fact that all of the functions $\psi_{\ell,k}$ are supported on $[0,1]$:

$$\begin{aligned}\psi_{\ell,k}(r) &= I^k\psi_{\ell,0}(r) = II^{k-1}\psi_{\ell,0}(r) = I\psi_{\ell,k-1}(r)\\ &= \int_r^\infty s\psi_{\ell,k-1}(s)ds = \int_r^1 s\psi_{\ell,k-1}(s)ds.\end{aligned}$$

□

Theorem 1 has a different flavor than the recursion formulae given in [8,9]. It is particularly helpful for our purposes since it sheds some light on the smoothness relations of the functions $\psi_{\ell,k}$. On the one hand we see that the functions in the family (2), which we used for our first set of numerical experiments, are *not* smoother versions of one another. Rather they are all a smoother version (in the sense of Theorem 1) of a function from a different family. For example, hierarchy (2) suggests that the C^6-function $\psi_{5,3}$ is a smoothed version of the C^4-function $\psi_{4,2}$. Theorem 1, however, implies that $\psi_{5,3}$ is a smoother version of $\psi_{5,2}$ (which is C^4, but strictly positive definite in $\mathbb{R}^5$ and not $\mathbb{R}^3$, as the other functions in (2)).

On the other hand Theorem 1 tells us how to select a family of functions which are obtained by repeated smoothing of one basic function. In particular, for the example in Table 2 we used the functions

$$\begin{aligned}
\psi_{7,2}(r) &\doteq (1-r)_+^9(80r^2+27r+3),\\
\psi_{7,3}(r) &\doteq (1-r)_+^{10}(320r^3+197r^2+50r+5),\\
\psi_{7,4}(r) &\doteq (1-r)_+^{11}(128r^4+121r^3+51r^2+11r+1),\\
\psi_{7,5}(r) &\doteq (1-r)_+^{12}(2048r^5+2697r^4+1644r^3+566r^2+108r+9),\\
\psi_{7,6}(r) &\doteq (1-r)_+^{13}(4096r^6+7059r^5+5751r^4+2782r^3+830r^2\\
&\quad +143r+11).
\end{aligned} \tag{6}$$

As a by-product we mention that the recursive formula of Theorem 1 can be used to obtain *explicit* formulae for the functions $\psi_{\ell,k}$ for fixed values of k.

Corollary 2. *The functions $\psi_{\ell,k}$, $k=1,2,3$, have the representation*

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

Corollary 2 gives explicit formulae for Wendland's C^2, C^4, and C^6 functions which can be made strictly positive definite in any $\mathbb{R}^d$ by choosing $\ell = \lfloor \frac{d}{2} \rfloor + k + 1$ (see [8]). Formulae for higher values of k (and thus higher smoothness) can also be computed with the help of symbolic manipulation software.

§4. The Functions $\psi_{\ell,k}$ and Convolution

In order to establish a connection with the general smoothing theory given in [2] we investigate the relation of the functions $\psi_{\ell,k}$ to convolution.

Theorem 3. *The functions $\psi_{\ell,k}$, $k \geq 1$, have the representation*

$$\psi_{\ell,k}(r) = r\psi_{\ell,k-1}(r) * \chi_{[r,\infty)}.$$

Proof: We show that the right-hand side is equal to $\psi_{\ell,k}(r)$. Due to the definition of convolution, the support of the characteristic function, and the compact support of $\psi_{\ell,k-1}$ we get

$$\begin{aligned} r\psi_{\ell,k-1}(r) * \chi_{[r,\infty)} &= \int_{-\infty}^{\infty} s\psi_{\ell,k-1}(s)\chi_{[r-s,\infty)} ds \\ &= \int_{r}^{\infty} s\psi_{\ell,k-1}(s) ds \\ &= \int_{r}^{1} s\psi_{\ell,k-1}(s) ds \\ &= \psi_{\ell,k}(r). \end{aligned}$$

$\square$

We now see that the smoothing S is defined via

$$Su(r) = ru(r) * \chi_{[r,\infty)} = \int_{r}^{\infty} su(s) ds,$$

and the functions in family (6) are given by

$$\psi_{7,k+2} = S^k \psi_{7,2}, \qquad k = 0, \ldots, 4.$$

According to the general theory in [2] the smoothing is required to be an approximate identity, and satisfy a Bernstein and Jackson type inequality. The verification of these properties is beyond the scope of this paper. It will be the subject of a future paper [3].

§5. Concluding Remarks

We have suggested two hierarchies of functions for the use in an efficient implementation of smoothing within the basic multilevel approximation algorithm. Simple numerical examples indicate the potential of both families. However, a rigorous proof of the properties of the smoothing operators, including a detailed analysis of the parameters involved, is deferred to [3].

For practical purposes a hierarchy as in (2) has the advantage that it is "open ended". By this we mean that one can start with some basic function ψ_{ℓ_0,k_0}, say, and then use $\{\psi_{\ell_0+i,k_0+i}\}$ as the smoother functions for the evaluation of the residuals in the multilevel algorithm as illustrated in the introductory section. The only real cost for implementing this scheme is the effort involved in precomputing the functions with a symbolic manipulation program such as MAPLE, and hardcoding them into the software.

On the other hand, it seems that a hierarchy such as (6) is better suited to the theoretical analysis required. Also, according to our simple numerical experiments, family (6) seems to be more effective than family (2). The drawback of (6) in practice is that one has to have an idea of how many iterations will be required, say K. Then the family of functions to be used is $\{\psi_{K+1,k}\}$. Again, all functions can be determined a priori symbolically, and then be coded into the software.

In any case, the construction of a suitable set of "smoother" basis functions seems to be preferential to the direct use of convolution for the smoothing at every level of the iteration (realized via fast Fourier transform, numerical quadrature, or similar means).

References

1. 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.
2. Fasshauer, G. E. and J. W. Jerome, Multistep approximation algorithms: Improved convergence rates through postconditioning with smoothing kernels, Advances in Comp. Math., to appear.
3. Fasshauer, G. E. and J. W. Jerome, A family of compactly supported radial basis functions for postconditioning of multilevel approximation algorithms, in preparation.
4. Floater, M. S. and A. Iske, Multistep scattered data interpolation using compactly supported radial basis functions, J. Comput. Applied Math. **73** (1996), 65–78.
5. Franke, C. and R. Schaback, Convergence orders of meshless collocation methods using radial basis functions, Advances in Comp. Math., to appear.
6. Narcowich, F. J., R. Schaback, and J. D. Ward, Multilevel interpolation and approximation, preprint, 1997.
7. 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.
8. Wendland, H., Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Advances in Comp. Math. **4** (1995), 389–396.
9. Wendland, H., Error estimates for interpolation by compactly supported radial basis functions of minimal degree, J. Approx. Theory, to appear.

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