SIAM J. SCI. COMPUT.
Vol. 24, No. 1, pp. 81 117
© 2002 Socie for Ind rial and Applied Ma hema ic

A MULTIRESOLUTION APPROACH TO REGULARIZATION OF SINGULAR OPERATORS AND FAST SUMMATION*

† AND †

Abstract.

Classically, integrals with nonintegrable singularities are given meaning by first defining the integral on test functions that vanish in a neighborhood of the singularity, then extending this definition to test functions that do not. Such a procedure is known as regularization. Typically, one considers the regularization that is “natural” in the sense that the sum of two ordinary kernels corresponds to the sum of their regularizations, the ordinary derivative of a kernel to the derivative of its regularization, and the product of the kernel with an infinitely differentiable function to the regularization of the product [10].

An effective way to arrive at the natural regularization is by analytic continuation with respect to a complex parameter λ. In this case, the original kernel is replaced by a family of kernels which are analytic with respect to λ in some domain, say Λ, on which the kernel is locally integrable. As an example, consider the integral

$$\int_0^\infty x$$

can be computed using FMM-type algorithms [11], [6]. Alternatively, the sum can be interpreted as an integral with a hypersingular kernel, regularized using our approach, then evaluated using a fast algorithm. This application is developed in section 6 of this paper.

Another application is in fluid mechanics where we encounter the projector onto spaces of divergence-free functions. The kernel of the projector is defined by

$$K_{ij}(x) = \delta_{ij}\delta(x) - C_n \quad \frac{\delta_{ij}}{x^{\ n}} - \frac{n\, x_i x_j}{x^{\ n+2}} \quad , \tag{3}$$

where C_n equals $1/2\pi$

[10]. The classical approach interprets a divergent integral as a functional, or generalized function, operating on a class of test functions. The origin of the mathematical treatment of generalized functions (distributions) goes back to the theory introduced by Schwartz (see, e.g., [16]). Such a functional, appropriately constructed, provides the definition for the classical regularization. We consider a natural regularization (see [10] or section 1). We note that divergent integrals involving functions with algebraic singularities are ubiquitous in physical applications. These functions increase as some power of $1/ x - x_0$, as x approaches the singular point x_0, and divergent integrals involving them serve as our main examples.

As a systematic method for regularizing such integrals, we may employ the method of analytic continuation. The main idea is to construct a family of generalized functions f_λ analytic with respect to a parameter λ over some open region Λ in the complex plane. If the functional can be extended analytically to a wider region, say Λ_1 , then we consider the analytic continuation of the functional as a definition of the generalized function f_λ for $\lambda \in \Lambda_1$.

We illustrate the main points with an example. Let us define $f_\lambda = x_+^\lambda$, where

$$x_+^\lambda = \begin{cases} x^\lambda, & x > 0, \\ 0, & x \leq 0. \end{cases}$$

For $\mathrm{Re}(\lambda) > -1$, this generalized function is defined by the convergent integral

$$(x_+^\lambda, \varphi) = \int_0^\infty x^\lambda \varphi(x)\, dx, \tag{5}$$

where $\varphi(x)$ belongs to the space of infinitely differentiable test functions with compact support. Splitting the integral in (5), we redefine the functional as

$$(x_+^\lambda, \varphi) = \int_0^1 x^\lambda [\varphi(x) - \quad x)\, dx,$$

in the first case and

$$(Tf)(x) = \int_0^\infty y^{-2m} \left(f(x-y) + f(x+y) - 2 \sum_{k=0}^{m-1} f \right.$$

We begin by assuming that integrals in (11) and (12) are convergent. With this assumption we derive a system of equations that allows us to compute the projection of the operator onto an MRA without evaluating any integrals. The computation requires knowledge only of the degree of homogeneity and values of the kernel $K(x)$ for large x. If integrals in (11) and (12) are not convergent, then we use this construction as a definition of multiresolution regularization.

3.1.1. Test functions. The subspaces V_j, of an MRA (see Appendix B) serve as the spaces of test functions. Let $\varphi(x)$ be the scaling function for the MRA. The functions

$$\varphi_{j,k}(x) = 2^{-j/2}\varphi(2^{-j}x - k) \quad k \in \mathbb{Z},$$

form an orthonormal basis for the subspace V_j.

We make extensive use of the two-scale difference equation (see Appendix B)

$$\varphi_{j,k}(x) = \sum_l h_l \varphi_{j-1,2k+l}(x)\,. \tag{13}$$

In what follows, we consider projection of a kernel onto the MRA and show that (13) leads to a two-scale difference equation for the coefficients of the projection which, for homogeneous kernels, relates coefficients on the same scale. This relationship, together with asymptotic behavior of the kernel $K(x)$ for $x \to \infty$, provides the means for obtaining the coefficients of the projection.

Formally applying the operator T to the basis function $\varphi_{j,k}$ we obtain the following proposition.

1. *An operator T with kernel homogeneous of degree α scales as*

$$(T\varphi_{j,k})(x) = 2^{-\alpha j}(T\varphi)_{j,k}(x)\,. \tag{14}$$

Proof. Rearranging (12) we have

$$(Tf)(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{K}(\xi) \int_{-\infty}^{\infty} f(y) e^{-i\xi(x-y)}\, dy\, d\xi\,.$$

Using this expression, we have

$$\begin{aligned}
(T\varphi_{j,k})(x) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{K}(\xi) \int_{-\infty}^{\infty} 2^{-j/2}\varphi(2^{-j}y - k) e^{-i\xi(x-y)}\, dy\, d\xi \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{K}(\xi) \int_{-\infty}^{\infty} 2^{j/2}\varphi(y) e^{-i2^j\xi(2^{-j}x-k-y)}\, dy\, d\xi \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{K}(2^{-j}\xi) \int_{-\infty}^{\infty} 2^{-j/2}\varphi(y) e^{-i\xi(2^{-j}x-k-y)}\, dy\, d\xi \\
&= 2^{-\alpha j}\, 2^{-j/2}(T\varphi)(2^{-j}x - k)\,,
\end{aligned}$$

where we used (10a). □

3.1.2. Multiresolution representation of the operator. Let us construct a sequence of operators T_j such that $T_j : V_j \to V_j$ and the kernel of T_j has the form

$$T_j(x, y) = \sum_m \sum_n t^j_{m-n}\varphi_{j,m}(x)\varphi_{j,n}(y)\,, \quad j \in \mathbb{Z}, \tag{15}$$

where t_n^j

where

$$a_m = \sum_k h_k h_{k+m}\,,$$

and $h_{k'}$ *are the coefficients in* (13).

Proof. First note that $\varphi(x) = \sum h_l \varphi_{-1,l}(x)$ from (13), and this, together with (14), implies

$$(T\varphi)(x) = \sum_l h_l (T\varphi_{-1,l})(x) = 2^{\alpha} \sum_l h_l (T\varphi)_{-1,l}(x)\,.$$

Using this expression we have

$$\begin{aligned} t_n &= \int \varphi(x-n)(T\varphi)(x)\,dx \\ &= \int \sum_k h_k \varphi_{-1,2n+k}(x)\, 2^{\alpha} \sum_l h_l (T\varphi)_{-1,l}(x)\,dx \\ &= 2^{\alpha} \sum_k \sum_l h_k h_l \int \varphi(2x-2n-k)(T\varphi)(2x-l)2\,dx \\ &= 2^{\alpha} \sum_m \end{aligned}$$

Multiresolution regularization is consistent with the classical definition. As previously noted, the classical regularization alters neither the degree of homogeneity of the kernel nor its asymptotic behavior at infinity, and we use only these two properties to uniquely determine the multiresolution regularization. The relationship between classical and multiresolution regularization is discussed more fully in section 3.4.

There are three steps in our construction.

Step 1. We assume the coefficients $t_{n'}$ in (18) are known for large n. Indeed, for sufficiently large n, the integrals defining t_n are convergent since the domain of integration does not contain the singularity of the kernel $K(x)$. As a practical matter, we assume the asymptotic condition

$$t_n = F(n) + O\ \frac{1}{\quad}$$

It follows that a generalized kernel homogeneous of (fixed) degree $k = 0, 1, 2, \ldots$ has the general form (see, e.g., [10])

$$K(x) = \frac{c_1}{x_+^{k+1}} + \frac{c_2}{x_-^{k+1}} + C\,\delta^{(k)}(x)\,, \tag{26}$$

where c_1, c_2, and C are constants and x_+ and x_- are defined in Appendix A.

If the kernel $K(x)$ is $\delta^{(k)}(x)$, then the operator T is simply kth derivative. This case has been considered in [4], where it was shown that if the wavelet basis has a sufficient number of vanishing moments, then the two-scale difference equation (25) reduces to

$$2^{-k}\mathbf{t} = A\mathbf{t}$$

plus an additional normalization condition. Thus, in this case, $\mathbf{t}$ is the eigenvector of the matrix A.

We also have the following proposition.

5 (see, e.g., [9]

If $\mathbf{t}$ satisfies (25), then we have

$$\begin{aligned} 2^{-\alpha}\mathbf{t} &= A\mathbf{t} + \mathbf{b} \\ \Longrightarrow \quad 2^{-\alpha}\mathbf{x}^T\mathbf{t} &= (\mathbf{x}^T A)\mathbf{t} + \mathbf{x}^T\mathbf{b} \\ \Longrightarrow \quad 0 &= \mathbf{x}^T\mathbf{b}, \end{aligned}$$

which implies (28). □

3.3. Asymptotic condition for integral operators. Let us establish the asymptotic condition (23). The coe□cients $t_{n'}$ are defined formally by

$$t_n = \quad \varphi(x - n)(T\varphi)(x)\,dx,$$

and thus

t

where we have used (30). Thus,

$$t_n - K(n) = \int \frac{K^{(2M)}(x_0)}{(2M)!}(x-n)^{2M}\Phi(x-n)\,dx,$$

where x_0 lies between x and n. The assertion (31) now follows upon demonstrating that the integral on the right, divided by $1/n^{2M+1+\alpha}$, is bounded for all sufficiently large n.

Differentiating (10b) repeatedly, we obtain

$$x^{2M}K^{(2M)}(x) = (\alpha+1)(\alpha+2)\cdots(\alpha+2M)K(x),$$

and combining this with the explicit form of $K(x)$ given in (26), we obtain

$$\frac{K^{(2M)}(x_0)}{(2M)!} = \frac{C}{x_0^{2M+1+\alpha}},$$

where C is a constant. Thus,

$$n^{2M+1+\alpha}\int \frac{K^{(2M)}(x_0)}{(2M)!}(x-n)^{2M}\Phi(x-n)\,dx = C\int \frac{(x-n)^{2M}}{(x_0 n^{-1})^{2M+1+\alpha}}\Phi(x-n)\,dx,$$

which is clearly bounded for n sufficiently large. □

3.4. Relationship between multiresolution and classical regularization. In section 3.3 we showed that the coefficients t_n, may be defined formally by

$$t_n = \int K(x)\Phi(x-n)\,dx. \tag{32}$$

However, it is necessary to use a regularization procedure to define t_n if this integral is divergent. We have achieved this using the multiresolution approach, but if the autocorrelation $\Phi(x$

where we have taken into account that Φ is an even function. Hence, as Proposition 8 shows, coefficients in (33) agree within prescribed accuracy with the coefficients used to initialize the regularization procedure in (23).

To see that coefficients $t_{n'}$ in (33) satisfy the two-scale difference equation (24), note that Φ satisfies

$$\Phi(x) = \sum_{m=-m_0}^{m_0} a_m \Phi(2x + m)$$

(see Appendix B) and, assuming that Φ is sufficiently differentiable, we also have

$$\Phi^{(2k)}(x) = 2^{2k} \sum_{m=-m_0}^{m_0} a_m \Phi^{(2k)}(2x + m) \qquad \text{for } k = 1$$

Let $y(x)$ denote the dual scaling function, constructed to satisfy

$$\beta(x)\, y(x-n)\, dx = \delta_{n,0}\,, \quad n \in \mathbb{Z}.$$

It follows that the coefficients $t_{n'}$ in (34) are defined formally by

$$t_n = \quad y(x-n)(Ty)(x)\, dx\,. \tag{35}$$

Since the dual scaling function is not compactly supported, $y(x-n)$ fails to vanish in a neighborhood of the singularity of the kernel of T, and the integral in (35) may fail to converge for all integers n. On the other hand, to start the regularization procedure, it is necessary to assign values to the coefficients t_n for large n in a consistent manner. In addition, the two-scale difference equation satisfied by y contains infinitely many fully coupled nonzero coefficients.

Rather than working with y directly, let us begin instead by considering the following expression, which is dual to (34):

$$T_0(x,y) = \quad \underset{j \quad k}{} \quad \tau_{j-k}\, y(x-j)\, y(y-k)\,, \tag{36}$$

where the coefficients $\tau_{n'}$ are defined formally by

$$\tau_n = \quad \beta(x-n)(T\beta)(x)\, dx\,. \tag{37}$$

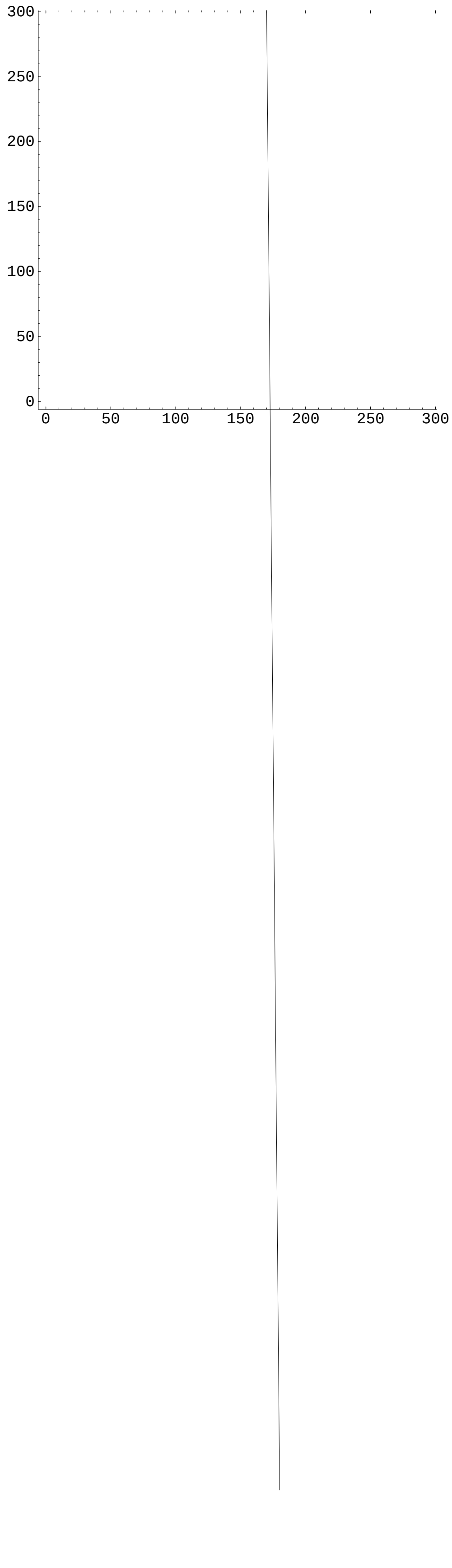
If the coefficients $\tau_{n'}$ are available, then, since the dual scaling function y can be expressed in terms of β, the coefficients $t_{n'}$ in (34) can be obtained directly from them.

Due to compact support of $\beta(x)$, integrals in (37) are convergent for all sufficiently large n, and we are able to compute coefficients τ_n directly from this integral expression to provide the necessary starting point for the regularization procedure. Equation (37) can be expressed as

$$\tau_n = \quad {}_{-M}^{\,M} \; B(x) K(x+n)\, dx\,,$$

Thus, the regularization procedure described in section 3 can be used, with (39) and (40) in place of (23) and (24), respectively, to obtain all coe□cients $\tau_{n'}$

(see Appendix A for definition of ξ_+ and ξ_-). Note that the derivative of the delta function also satisfies (43) but, as explained in section 3.2.1, we exclude this case from consideration. If $\alpha = 0, 1, 2, \ldots$, then the case of $c_1 = (i)^{\alpha+1}\pi/\alpha! = -c_2$ in (44)


300
250
200
150
100
50
0
0
50
100
150
200
250
300

6. Fast summation of discrete sums. In this section we develop an application, namely, a method for fast summation of discrete sums of the form

$$g_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} K(\mathrm{x}_i - \mathrm{x}_j) f_j, \tag{48}$$

where $\mathrm{x}_i \in \mathbb{R}^n$ are particle locations, and f_i is the charge carried by the ith particle. The kernel $K(\mathrm{x})$ is a homogeneous function which describes interparticle interactions. For vector-valued kernels we apply the algorithm in each index.

Particle models are frequently encountered in the computer study of physical systems. Among numerous examples are N-body simulations in astrophysics and vortex methods in fluid mechanics. In many such models evaluation of a discrete sum, which accounts for pairwise interaction between particles, is the most expensive part of the computation. To account for the pairwise interactions directly requires $O(N^2)$ operations for an N-particle system. In this section, we present an algorithm to accomplish this in $O(N + N \log N)$ operations.

The basic computational problem in particle models may be viewed as that of computing the value, at each particle location, of the potential field generated by the particle ensemble, while excluding the self-interaction which is generally infinite. To provide background, we mention two algorithms, namely, the fast multipole method (FMM) and the method of local corrections (MLC). The FMM (see, e.g., [11]) has been highly successful in constructing fast algorithms for a variety of summation problems and incorporates several ideas which are common to such algorithms. The MLC [3] was introduced as a vortex method for problems in fluid mechanics, though the main ideas are applicable in a wider context.

After discussing these two algorithms, we present an algorithm based on multiresolution regularization (section 3), which we compare to FMM and MLC. Our approach may be viewed as similar to either of these, depending on how we choose to apply the multiresolution kernel. Choosing Fourier transform methods produces an algorithm similar to MLC, but we can also exploit the wavelet decomposition and the nonstandard form [5], which produces an algorithm similar to FMM.

Remark 7. We note that "modern" FMM (see, e.g., [6]) uses approximations with exponentials, which significantly improves its efficiency. The incorporation of similar approximations into our algorithm is in progress and will be reported elsewhere.

For simplicity, we describe the earliest version of FMM. In this approach, a sum of the form (48) is expanded as a Laurent series, or multipole expansion. At points distant from the particle ensemble, the expansion takes the form of a rapidly converging power series, and this far-field potential is well approximated by only a few terms of the expansion. If the number of terms required to achieve the desired accuracy is less than the number of particles, then evaluating the multipole expansion requires less effort than evaluating the sum (48), and a significant increase in computational speed may be realized. This method of computing the far-field potential is the basic mechanism for gaining computational efficiency in the FMM.

To exploit this mechanism, a hierarchical subdivision of space into boxes on several scales is constructed, which induces a subdivision of the particle ensemble into subcollections. By introducing several scales into the model, computation of the interaction between different subcollections can be performed on a scale at which they are well separated, which allows for use of the far-field expansions.

Beginning with the finest scale, a multipole expansion is constructed for each box at each level of the hierarchy, which represents the far-field potential due to the particles in the box. Expansions on coarser levels are obtained by merging expansions at the next finer level. After completing this step, the far-field interactions can be computed for each box.

Beginning with the coarsest scale, interactions between well-separated boxes are computed using the multipole expansions. These contributions are accumulated in the form of a multipole expansion for each box, which is then translated to the box subdivisions on the next finer scale. This procedure is repeated until a multipole expansion has been constructed for each box in the hierarchy, which represents the far-field potential due to the particle subcollections in all well-separated (exterior) boxes.

The final step in the FMM is to compute all near-field particle interactions directly. Since the near-field potential at each particle location involves only a few neighboring particles, the number of operations required for this step is a constant times N, where N is the number of particles and the constant is small relative to N.

The MLC also seeks to evaluate a sum of the form (48), which represents the velocity field induced by an ensemble of point vortices. The velocity field of a point vortex becomes unbounded near the vortex center but is smooth elsewhere. Thus, in MLC as in FMM, the basic strategy is to approximate the velocity field of a point vortex at distant points by polynomials, while using an explicit formula for points of the field near the vortex center.

MLC begins by constructing an approximation to the velocity field at each point of an equally spaced grid overlaying the computational domain. This construction involves solution of a discretized Laplace equation and is extended from the grid to the vortex centers by a polynomial interpolating function. The number of operations needed for constructing and evaluating the approximate velocity field is proportional to $M \log M$, where M is the number of grid points.

Each point vortex is approximated by a radially symmetric function with finite support, called a "vortex blob," and the approximation to the velocity field of a vortex blob agrees closely with the actual velocity at points sufficiently far from the vortex center, but diverges from the correct velocity near the center. This implies that the approximation to the total velocity field, evaluated at a vortex center, contains the correct contribution from distant vortices, but the contribution from nearby vortices is in error. Since the distance between vortex centers is measured relative to the grid spacing, one can always rescale to make all vortex centers well separated, but this is generally inefficient. A more efficient strategy is to correct the initial approximation at each vortex center to remove errors due to nearby vortices.

To correct the initial approximations, the MLC first computes the contribution to the approximate velocity field due to nearby vortices, then subtracts this quantity and adds the correct contribution obtained from an explicit formula. As in the FMM, this last step involves only a few nearby vortices for each vortex center, and thus the number of operations required for this step is a constant times N, where N is the total number of vortex blobs and the constant is small relative to N.

Remark 8. We note that MLC does not obtain an explicit representation of the correction operator, which is done in our approach (see section 6.4). Such explicit representations are sometimes useful, especially if the problem is not restricted to evaluating sums.

6.1. Multiresolution algorithm for fast summation. We use the methods for regularizing singular and hypersingular operators described above to develop an algorithm for fast computation of the vector $g_1, \ldots, g_{N'}$, where g_i is defined by (48).

Due to the nature of the kernel $K(x)$ in (48), the potential field of a particle is easily approximated by smooth functions at points sufficiently distant from the particle locations, a situation similar to that encountered in our discussion of FMM and MLC above. The main difference in our approach is that instead of using polynomials to approximate the far-field of a point charge (or point vortex), we use the multiresolution regularization of the kernel K on the scale j (see section 3.2), denoted by $T_j(x, y)$.

As in the MLC, the multiresolution algorithm consists of two steps: an approximation step and a correction step. In the approximation step, we replace the kernel $K(x-y)$ in (48) by its multiresolution regularization $T_j(x, y)$ and perform the summation. We choose to carry out the summation using an FFT, and thus the number of operations required for this step is proportional to $M \log M$, where M is a grid-size determined by the scale of the projection. Alternatively, this operation could be carried out using the nonstandard form of the kernel T_j (see [5]).

It is shown in Appendix C that the kernel $K(x-y)$ is well approximated by $T_j(x, y)$ if and only if the distance $|x-y|$ is sufficiently large. Thus, in replacing K by T_j in (48), we have introduced significant errors only for interactions between pairs of particles that are not well separated, while interactions between well-separated particles have been computed to within the desired precision (analogous to the situation with MLC).

We can always choose a scale of resolution so fine that all pairs of particles in the ensemble are well separated, since a multiresolution regularization is easily rescaled (see (16)). However, choosing ever finer scales is generally not an efficient strategy, because the number of grid points eventually becomes large enough to degrade performance. As in the MLC we perform a second step to correct the errors in the initial approximation due to particles that are close together. For each particle, we compute the contribution to the initial approximation due to nearby particles, then subtract this quantity and add the correct contribution obtained using the original kernel K. In contrast to the MLC, we obtain an explicit representation of the correction operator. For each particle, this step involves only a few nearby particles, and thus the computational cost of this step is a constant times N, where N is the total number of particles and the constant is small relative to N.

As already mentioned, we could also apply the multiresolution regularization via the nonstandard form [5], although we do not demonstrate this in the present work. In this case we would not need a "correction step," and the algorithm would more closely resemble the FMM.

Remark 9. We begin by describing the fast summation algorithm in a one-dimensional setting, and then discuss the higher-dimensional implementation. One-dimensional formulas are readily transformed into higher-dimensional formulas, using the tensor product construction (see Appendix B), by simply treating all real scalar variables as vectors and integer indices as multi-indices.

Although the derivation does not change in two dimensions there is one important additional feature used: for a wide class of problems, the correction operator has a low separation rank (up to chosen precision). This allows us to use singular value decomposition of the coefficient matrix to significantly increase the speed of evaluating the correction operator.

6.2. "Reverse discretization" of the sum. To make use of the regularization technique developed in section 3, we interpret (48) as an integral operator by interpreting the numbers $\{g_{m'}\}$ as values of a function $g(x)$ defined by

$$g(x) = \int K(x-y) f(y)\, dy. \tag{49}$$

It is therefore necessary to correct only those contributions due to particles that are

6.1

	apr	cor	alg	dir	E_∞

6.2

To approximate the kernel $K(x-x', y-y')$, we construct the multiresolution regularization of K,

$$T_j(x,y,x',y') = \sum_{k,l} \varphi_{jk}(x)\varphi_{jl}(y) \sum_{k',l'} t^j_{k-k',l-l'}\varphi_{jk'}(x')\varphi_{jl'}(y'), \tag{60}$$

as described in section 3, using the two-dimensional two-scale difference equation (22) together with known asymptotic behavior of $K(x,y)$ as $\max(|x|,|y|) \to \infty$. Analogous to the one-dimensional estimate (54), there exists a constant B_j such that, for each $\epsilon > 0$, we have

$$|K(x-x', y-y') - T_j(x,y,x',y')| < \epsilon \tag{61}$$

if $\max(|x-x'|, |y-y'|) > B_j$ (see Appendix C).

6.6.1. **Approximation step.** The initial approximations have the form

$$g_{j,m} = \sum_{n=1}^{N} T_j(x_m, y_m, x_n, y_n) f_n, \qquad m = 0, 1, \ldots, N. \tag{62}$$

Rearranging the sums, we obtain

$$g_{j,m} = \sum_{k,l} \hat{s}^j_{k,l}\varphi_{j,k}(x_m)\varphi_{j,l}(y_m), \tag{63}$$

where

$$\hat{s}^j_{k,l} = \sum_{k',l'} t^j_{k-k',l-l'} s^j_{k',l'} \tag{64}$$

and

$$s^j_{k',l'} = \sum_{n=1}^{N} f_n \varphi_{j,k'}(x_n)\varphi_{j,l'}(y_n). \tag{65}$$

The operation indicated in (64) is accomplished using a two-dimensional FFT.

6.6.2. **Correction step.** As explained above it is necessary to correct errors in the initial approximation due to particles that are too close together at the chosen scale of resolution j. To accomplish this, we subtract the erroneous contribution from the nearby particles and add the correct contribution. Thus, it is necessary to evaluate the multiresolution kernel $T_j(x,y,x',y')$ in the region defined by $\max(|x-x'|, |y-y'|) \le B_j$. For this purpose we utilize the trigonometric expansion of T_j, as obtained in Appendix D. In two dimensions, the (already truncated) form is

$$\begin{aligned} &2^{j(\alpha+2)} T_j(x_m, y_m, x_n, y_n) \\ &= I_{0,0} + 2\sum_{k=1}^{3} I_{k,0}\cos(2^{-j}k\pi(x_m+x_n)) + 2\sum_{l=1}^{3} I_{0,l}\cos(2^{-j}l\pi(y_m+y_n)) \\ &\quad + 4\sum_{k=1}^{3}\sum_{l=1}^{3} I_{k,l}\cos(2^{-j}k\pi(x_m+x_n))\cos(2^{-j}l\pi(y_m+y_n)), \end{aligned} \tag{66}$$

where

$$I_{k,l} = I_{k,l} \ 2^{-}$$

expression (76b), which converges in the strip $-2m-1 < \mathrm{Re}(\lambda) < -2m+1$. In particular, we have

$$(x^{-2m-1}, \varphi) = \int_0^\infty x^{-2m-1} \left[\varphi(x) - \varphi(-x) - 2\sum_{k=1}^{m} \frac{\varphi^{(2k-1)}(0)}{(2k-1)!} x^{2k-1} \right] dx\,,$$

$$(x^{-2m}, \varphi) = \int_0^\infty x^{-2m} \left[\varphi(x) + \varphi(-x) - 2\sum_{k=1}^{m} \frac{\varphi^{(2k-2)}(0)}{(2k-2)!} x^{2k-2} \right] dx\,.$$

B.2. Tensor product construction. To construct an MRA for $L^2(\mathbb{R}^n)$, the simplest method is to form the tensor product of an MRA for $L^2(\mathbb{R})$ (see, e.g., [9], [15]). For example, if $\varphi(x-k)$ $k \in \mathbb{Z}$ is a basis for V_0

where M is the number of vanishing moments in the MRA. (This result depends on orthonormality of functions $\varphi(x-k)$.) Function Φ satisfies a two-scale difference equation,

$$\Phi(x) = \sum_m a_m \Phi(2x-m), \tag{90}$$

where the coefficients a_m

The coefficients $t^j_{m-n'}$ in (95) are defined formally by

$$t^j_{m-n} = \quad K(u-v)\varphi_{j,m}(u)\varphi_{j,n}(v)\,du\,dv\,, \tag{98}$$

and from (97) it follows that the integral in (98) converges if $m-n \;> 2s$. It follows also that, for a given (x, y), the summation in (95) involves a finite number of terms. Let us impose the following condition:

$$x-y \;\geq 2^j(4s) + \delta\,, \qquad \delta > 0\,. \tag{99}$$

Then all coefficients t^j_{m-n} involved in the summation are defined by a convergent integral in (98).

For the remainder of the proof let the point (x, y) be fixed and chosen such that where ξ is between u and x, and η is between v and y, we obtain

$$T_j(x,y) - K(x-y) \le C_1 \sup \frac{K^{(M)}(\xi-\eta)}{M!},$$

where

$$C_1 = \quad [(u-v)-(x-y)]^M P_j(x,u)P_j($$

where

$$\Phi_n(z) = 1$$

[ ] C. A