Periodic boundary conditions and the Fast Multipole Method[2]

Matt Challacombe[1], Chris White, and Martin Head-Gordon
Lawrence Berkeley National Laboratory,
1 Cyclotron Road, Berkeley, California 94720
and
Department of Chemistry, University of California, Berkeley
410 Latimer Hall, Berkeley, California 94720-1460
(July 5, 1997)

The seminal work of Nijboer and De Wette [Physica, **23**, 309 (1957)] enables the calculation of lattice sums of spherical harmonics, but has long been overlooked. In this article, their central result is recast in a simplified form suitable for modern multipole algorithms that employ the solid harmonics. This formulation makes possible the imposition of periodic boundary conditions within modern versions of the Fast Multipole Method, and other fast N-Body methods. The distinction between the extrinsic values obtained with the lattice sums $\mathcal{M}$ of the multipole interaction tensors, and the intrinsic values associated with Taylor's expansion of the Ewald formulae, is made. The central constants, $\mathcal{M}$, are computed to 32 digit accuracy using extended precision arithmetic. Timings and corresponding errors obtained with a periodic version of the Fast Multipole Method are presented for particle numbers spanning $[10^3, 10^6]$, and a range of expansion orders. A qualitative comparison is made between the present implementation, other periodic versions of the Fast Multipole Method, and fast Ewald methods.

I. INTRODUCTION

Periodic boundary conditions are often employed in the simulation of liquids and solids, where long range Coulombic effects are important. In this approach, charges in the simulation cell interact with each other and with those in the cell's periodic replicas. As the simulation cell becomes large, properties of the bulk may be computed reliably.

In a periodic simulation, central quantities are the electrostatic potential

$$\Phi(\mathbf{R}) = \sum_{\mathbf{n}} \sum_{j=1}^{N} \frac{q_j}{|\mathbf{R} - \mathbf{r}_j + \mathbf{n}|} \tag{1}$$

and the electrostatic energy

$$E = \frac{1}{2} \sum_{\mathbf{n}} {\sum_{ij}}' \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}|} \,, \tag{2}$$

where a unit $(1 \times 1 \times 1)$ cell of cubic symmetry has been, and will continue to be, used. In Eq. (2), the primed sum indicates that for $\mathbf{n} = \mathbf{0}$ the singular term involving $i = j$ is omitted. Also, the sum over $\mathbf{n}$ is in principle infinite, requiring charge neutrality of the simulation cell to arrive at a meaningful result.

Although slowly convergent and dependent on the manner of summation, it is possible to evaluate Eqs. (1) or (2) directly to a precision of ~ 4 digits in a few cpu hours. However, the convergence and the limiting value of these direct lattice sums depends on the way in which the sum is carried out [1–3], *e.g.* in spherical shells, ellipsoidal shells, slabwize, etc. This is because, even in the limit of an infinite sum, there exists a long range contribution from the surface that is shape dependent.

Alternatively, Eqs. (1) or (2) may be summed with Ewald methods [4,5], which are based on the identity

$$\frac{1}{|\mathbf{r}|} = \frac{\mathrm{erfc}(\beta|\mathbf{r}|)}{|\mathbf{r}|} + \frac{\mathrm{erf}(\beta|\mathbf{r}|)}{|\mathbf{r}|} \,. \tag{3}$$

Sums involving the first term on the RHS of Eq. (3) are localized in real space. On the other hand, sums involving the second term in Eq. (3) are long-ranged and smooth, and therefore localize well in reciprocal (Fourier) space. With the proper choice of β, the real and reciprocal space sums are rapidly convergent.

The potential and electrostatic energy obtained by Ewald summation are *intrinsic* properties, and differ from the *extrinsic* values obtained from direct summation by a shape dependent polarization correction [1–3,6]. Properties computed with the Ewald sum are shape independent (hence intrinsic), as they correspond to conducting "tin foil" boundary conditions that screen off all surface effects. The polarization corrections that relate extrinsic to intrinsic properties are well known analytic forms [3,6]. While the direct sum corresponds to a dielectric constant $\epsilon = 1$, and the Ewald sum to $\epsilon = \infty$, obtaining results for other values of ϵ involves a simple correction [4]. Thus, the imposition of boundary conditions is independent of the method, and the choice of direct or Ewald summation reduces to one of computational efficiency and numerical accuracy.

Algorithms for performing Ewald sums are well understood and have achieved a high level of sophistication as exemplified by a variety of fast Ewald algorithms [7–10] that are based on well developed 3D FFTs, encumbering only an $\mathcal{O}(N \log N)$ cpu dependence. Methods based on Fast Multipole Methods (FMMs) exist for computing the direct sums [11–14] and the Ewald sums [15–18] in $\mathcal{O}(N)$ cpu time, but are less well developed than the fast Ewald algorithms, due perhaps to their recent emergence, a lack of historical focus in the literature, and their greater difficulty to implement.

---

[1]E-Mail: MChalla@T12.LANL.Gov
[2]To appear in the Journal of Chemical Physics.

In this paper, our goal is to highlight the seminal work of Nijboer and De Wette [19–21], to recast their central result in a simplified form for use with modern multipole methods based on the solid harmonics [22–25], and to attempt a qualitative comparison with alternative state-of-the-art methods. The paper is organized as follows: In Section II, the multipole approach to direct lattice sums is briefly reviewed, and a modification of the NDW method compatible with modern multipole methods is developed and clarified. Then in Section III, our implementation of the NDW method is discussed, its interface with a modern FMM algorithm is outlined, and results are presented for particle numbers and levels of accuracy that span multiple orders. Next, we compare our results with alternative versions of the periodic FMM as well as with fast Ewald methods based on the 3D FFT. Finally, we conclude.

II. THE MULTIPOLE APPROACH TO DIRECT LATTICE SUMS

One approach to obtaining a multipole representation of the lattice potential is to simply expand the corresponding Ewald expression in a Cartesian Taylor's series and then perform the resulting real and reciprocal space sums. This approach may be traced back at least to the work of De Leeuw *et al.* [3], and has since been used in conjunction with Cartesian generalizations of the FMM by Hinsen [15], Ding *et al.* [16], and Shimada *et al.* [17,18]. Although computation of the order p Cartesian multipole interaction tensor may be carried out efficiently in $\mathcal{O}(p^4)$ cpu time using recursion relations [26], the solid harmonic interaction tensor [22–25] may be computed in $\mathcal{O}(p^2)$ cpu time. We have compared highly optimized implementations of both expansion types and found that expansions in solid harmonics are computationally more efficient than Cartesian expansions for $p > 2$ [27].

Recently, Delly [28] has proposed transforming from the effective Cartesian interaction tensors (resulting from expansion of the Ewald potential) to spherical ones via numerical quadrature on the unit sphere. It is also possible to express the Cartesian tensors in an irreducible (spherical harmonic like) representation using the method of Hinsen [15,29]. Yet another economized approach to approximation of the Ewald potential involves expansion in the Kubic harmonics, which are solutions of Lapace's equation with cubic symmetry [30].

In the multipolar approach to direct summation, $|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}|^{-1}$ is developed in a multipole series, allowing the multipole interaction tensor to be summed over all $\mathbf{n}$, outside the loop over i and j. The resulting effective tensor incorporates the interaction of the simulation cell with the lattice, and is a universal constant of the cell shape and summation order.

In 1957, Nijboer and De Wette (NDW) [19] introduced an elegant and general method for computing the lattice sums of spherical harmonic interaction tensors using Ewald-like partitioning and Parseval's theorem. A fundamental difference between the NDW result and the aforementioned methods [3,15–18,28,30] is that they are expansions of the Ewald potential, and yield intrinsic properties, while the NDW result is equivalent to direct summation and yields extrinsic properties.

Unfortunately, the importance of NDW's work has largely gone unrecognized. In 1991, Schmidt and Lee (SL) [11] incorporated the NDW result into the FMM, but presented an incorrect formulae (Eq. 11) and failed to give a corresponding derivation or citation. In 1995, Singer [12] pointed out the connection of the SL equation to the work of De Wette and Nijboer, but this article, published in a mathematical journal, has received little attention. To underscore the importance of bringing the NDW result to light, we note that in a recent review of fast methods for periodic systems [5], NDW is cited but not referenced in the text. Moreover, Figueirido *et al.* [14], noting the SL error, recently re-derived the NDW result for the unit cube using the method of De Leeuw *et al.* [3], but do not cite NDW. In addition to giving credit where credit is due, the NDW result deserves recognition because it applies to the simple perfect (Bravis) lattices, not just to the unit cube considered here and in Refs. [12,14].

In this section, the NDW method is simplified for the case of the unit cube, and presented in a form suitable for use with modern multipole technologies [22–25] that employ solid harmonics.

A. Lattice sums of the interaction tensor

In the notation of White and Head-Gordon [22], the bipolar multipole expansion is

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \sum_{l_p=0}^{\infty}(-1)^{l_p}\sum_{l_q=0}^{\infty}\sum_{m_p=-l_p}^{l_p}\sum_{m_q=-l_q}^{l_q}\Big(O_{l_p}^{m_p}[\mathbf{r}-\mathbf{P}]\, M_{l_p+l_q}^{m_p+m_q}[\mathbf{P}-\mathbf{Q}]\, O_{l_q}^{m_q}[\mathbf{r}'-\mathbf{Q}]\Big)\,, \tag{4}$$

where the regular

$$O_l^m[\mathbf{r}] = |\mathbf{r}|^l\, \tilde{P}_l^m(\cos\theta_{\mathbf{r}})e^{-im\phi_{\mathbf{r}}} \tag{5}$$

and irregular

$$M_l^m[\mathbf{R}] = |\mathbf{R}|^{-(l+1)}\;\;\tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{R}})e^{im\phi_{\mathbf{R}}} \tag{6}$$

solid harmonics are defined in terms of the scaled associated Legendre polynomials

$$\tilde{P}_l^m(x) = \frac{1}{(l+m)!}\,P_l^m(x) \tag{7}$$

and

$$\tilde{\tilde{P}}_l^m(x) = (l-m)!\,P_l^m(x)\,. \tag{8}$$

A key feature of these polynomials, exploited by modern multipole methods, is that they may be computed efficiently with recurrence relations [22–25,31].

The bipolar expansion, Eq. 4, is equivalent to the result of Carlson and Rushbrooke [32], and is absolutely convergent when

$$|\mathbf{r}-\mathbf{P}| + |\mathbf{r}'-\mathbf{Q}| < (\sqrt{2}-1)|\mathbf{P}-\mathbf{Q}|\,. \tag{9}$$

Keeping this requirement in mind, Eq. (4) can be substituted into the sum $\sum_{\mathbf{n}}|\mathbf{r}_i-\mathbf{r}_j+\mathbf{n}|^{-1}$ occurring in Eq. (1)

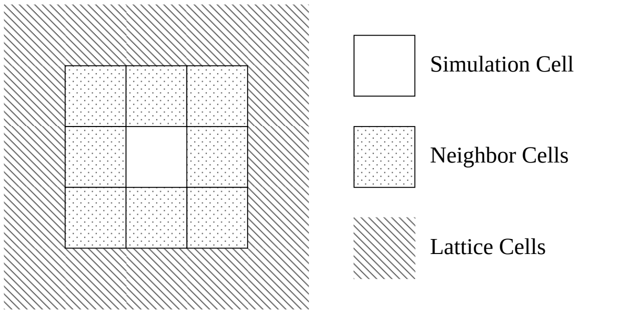


FIG. 1. Cell partitioning in lattice sums of the multipole interaction tensor for ws=1. In three dimensions, 26 neighboring cells insure convergence of the multipole expansion between charges in the simulation cell and the lattice cells.

or Eq. (2). Setting $\mathbf{P}=0$, and taking $\mathbf{Q}=\mathbf{n}$, the sum over $\mathbf{n}$ can be brought inside the tensor contraction to yield, in the case of Eq. (2),

$$E=\frac{1}{2}\sum_{l_p=0}^{p}(-1)^{l_p}\sum_{l_q=0}^{p}\sum_{m_p=-l_p}^{l_p}\sum_{m_q=-l_q}^{l_q}O_{l_p}^{m_p}\mathcal{M}_{l_p+l_q}^{m_p+m_q}O_{l_q}^{m_q} +\frac{1}{2}\sum_{ij}{}' \frac{q_i q_j}{|\mathbf{r}_i-\mathbf{r}_j|}\ , \tag{10}$$

where the effective multipole interaction tensors $\mathcal{M}_l^m = \sum_{\mathbf{n}\neq 0} M_l^m[\mathbf{n}]$ are universal constants of the cell shape and order of summation, which may be computed once and for all by the NDW method, direct summation [13], or renormalization methods [33]. Also in Eq. (10), the tensors O without an argument are understood to be multipole moments of the unit cell, $O_l^m=\sum_i q_i O_l^m[\mathbf{r}_i]$.

Because the bipolar expansion will converge only when Eq. (9) is satisfied, we consider effective multipole interaction tensors of the form

$$\mathcal{M}_l^m(\mathrm{ws})=\sum_{|\mathrm{ws}|<n_x}\sum_{|\mathrm{ws}|<n_y}\sum_{|\mathrm{ws}|<n_z} M_l^m[\mathbf{n}] \tag{11}$$

where ws is a well separated index that controls the number of neighbor boxes (26 for ws=1, 124 for ws=2, etc.), and hence convergence of the multipole expansion, in a way that is completely analogous to non-periodic versions of the FMM. Figure 1 shows a two-dimensional schematic of this partitioning. It then remains to compute the particle-particle interactions within the simulation cell and the neighboring cells by conventional or fast methods [34].

B. The Nijboer-De Wette algorithm applied to lattice sums of the irregular solid harmonics

The NDW algorithm is based on the Ewald-like partition,

$$\frac{1}{r^{l+1}}=\mathcal{G}_l(\beta,r)+\mathcal{F}_l(\beta,r)\ , \tag{12}$$

where

$$\mathcal{G}_l(\beta,r)=\frac{\Gamma\left(l+\frac{1}{2},\beta^2r^2\right)}{\Gamma\left(l+\frac{1}{2}\right)r^{l+1}}\ , \tag{13}$$

$$\mathcal{F}_l(\beta,r)=\frac{\gamma\left(l+\frac{1}{2},\beta^2r^2\right)}{\Gamma\left(l+\frac{1}{2}\right)r^{l+1}}\ , \tag{14}$$

$\gamma(a,x)$ is the incomplete gamma function, $\Gamma(a,x)=\Gamma(a)-\gamma(a,x)$ is its complement [35], and β is a parameter controlling convergence of the method. As in the Ewald method, each lattice sum can be partitioned into a rapidly convergent real space term

$$A_l^m(\mathrm{ws},\beta)= \sum_{|\mathrm{ws}|<n_x}\sum_{|\mathrm{ws}|<n_y}\sum_{|\mathrm{ws}|<n_z}\tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{n}})\,e^{im\phi_{\mathbf{n}}}\mathcal{G}_l(\beta,|\mathbf{n}|)\ , \tag{15}$$

and a smooth term

$$B_l^m(\mathrm{ws},\beta)= \sum_{|\mathrm{ws}|<n_x}\sum_{|\mathrm{ws}|<n_y}\sum_{|\mathrm{ws}|<n_z}\tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{n}})\,e^{im\phi_{\mathbf{n}}}\mathcal{F}_l(\beta,|\mathbf{n}|)\ . \tag{16}$$

As shown in Appendix A, this smooth term can be recast in reciprocal space as the rapidly convergent sum

$$\begin{aligned}\mathcal{B}_l^m(\mathrm{ws},\beta)=&\frac{i^l(2\pi)^{3/2}}{2^{l-1/2}\Gamma(l+1/2)}\\ &\times\sum_{\mathbf{n}} n^{l-2}e^{-n^2\pi^2/\beta^2}\ \tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{n}})e^{im\phi_{\mathbf{n}}}\\ &-\sum_{n_x=-\mathrm{ws}}^{\mathrm{ws}}\sum_{n_y=-\mathrm{ws}}^{\mathrm{ws}}\sum_{n_z=-\mathrm{ws}}^{\mathrm{ws}}\tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{n}})\,e^{im\phi_{\mathbf{n}}}\mathcal{F}_l(\beta,|\mathbf{n}|)\ .\end{aligned} \tag{17}$$

The effective interaction tensor elements may then be computed as $\mathcal{M}_l^m(\mathrm{ws})=\mathcal{A}_l^m(\mathrm{ws},\beta)+\mathcal{B}_l^m(\mathrm{ws},\beta)$. As discussed in Section III A, it is relatively straight forward to compute the low order elements, but obtaining accurate values for high orders requires more care.

C. Symmetry and convergence

As can be seen by inspection of Eq. (17), there are clear divergences in $\mathcal{B}$ that arise when $l<2$. In the case of monopole-monopole and monopole-dipole interactions, this divergence may be safely ignored as $O_0^0=0$ for charge neutral systems. Less obvious is the dipole-dipole interaction. In particular, the term $\mathcal{M}_2^m$ depends conditionally on the manner of summation, or equivalently on the way in which $\mathbf{k}$ approaches zero [36,20]. If the sum is taken slabwize, the elements $\mathcal{M}_2^m$ are finite [20]. However, if summed spherically, the $\mathcal{M}_l^m$ are real and exist only for l even, $l\geq 4$, and m a multiple of 4. The $\mathcal{M}$ are quite sparse, involving only $\mathcal{O}(l/2+l^2/16)$ non-redundant terms for an order l tensor. In contrast, there are $\mathcal{O}(3l+l^2)$ non-redundant terms that must be considered for an order l tensor M.

D. Ewald summation techniques in perspective

The NDW approach has sometimes been referred to as the Ewald method, which is a source of confusion in the literature. First, the Ewald method leads to intrinsic values while the

NDW method does not. Secondly, the straightforward generalization of the Ewald method that obtains from splitting the Laplace transform of $r^{-(l+1)}$ yields the identity (compare Eqs. (13) and (14))

$$\frac{1}{r^{l+1}} = \frac{\Gamma\left(\frac{l+1}{2}, \beta^2 r^2\right)}{\Gamma\left(\frac{l+1}{2}\right) r^{l+1}} + \frac{\gamma\left(\frac{l+1}{2}, \beta^2 r^2\right)}{\Gamma\left(\frac{l+1}{2}\right) r^{l+1}} , \tag{18}$$

which does not lead to a closed form for the computation of $\mathcal{M}$. Thus, the NDW result is fundamentally different from the Ewald method, although the computed properties may be simply inter-converted.

E. Intrinsic properties from direct lattice sums

As the NDW method is equivalent to spherically ordered direct lattice sums (ss), converting to the intrinsic Ewald energy E_{Ew} simply involves the well known [3,6] extrinsic correction

$$E_{\text{Ew}} = E_{\text{ss}} - \frac{2\pi}{3}\mathbf{d}\cdot\mathbf{d} , \tag{19}$$

where $\mathbf{d} = \sum_{\mathbf{i}} \mathbf{q_i r_i}$ is the cell dipole moment. Perhaps less well known is the relationship [1,6] between the intrinsic potential and the spherically summed potential

$$\Phi_{\text{Ew}}(\mathbf{R}) = \Phi_{\text{ss}}(\mathbf{R}) - \frac{4\pi}{3}(\mathbf{R}-\mathbf{R}_0)\cdot\mathbf{d} + \frac{2\pi}{3}Q , \tag{20}$$

where $\mathbf{R}_0$ is the origin and $Q = \sum_i q_i \mathbf{r}_i \cdot \mathbf{r}_i$ is the trace of the Cartesian quadrupole tensor. The extrinsic correction to the force may be found simply by differentiating Eq. (19).

III. RESULTS

A. Computation of the $\mathcal{M}$

With an appropriate choice of $\beta \sim \sqrt{\pi}$, the effective multipole interaction tensor $\mathcal{M}_l^m(\text{ws}) = \mathcal{A}_l^m(\beta,\text{ws}) + \mathcal{B}_l^m(\beta,\text{ws})$ may be computed in a finite amount of time as terms in both of the potentially infinite sums occurring in Eq. (15) and Eq. (17) die off as the Gaussians $n^l e^{-\beta^2 n^2}$ and $n^{l-2} e^{-n^2\pi^2/\beta^2}$ respectively. The only potential complication is computation of γ and Γ.

One approach [11,12,14] is to compute the incomplete gamma function with upward recurrence of the relation

$$\Gamma(m+1,x) = m\Gamma(m,x) + x^m e^{-x} , \tag{21}$$

which can be started with

$$\Gamma(1/2,x) = \sqrt{\pi}\,\text{erfc}(x) , \tag{22}$$

where the complimentary error function, erfc, is typically an intrinsic function in the Fortran and C languages. We will refer to this approach, implemented in double precision IBM xlf Fortran and starting with the DERFC function, as the standard method. Note that downward recurrence relations for the incomplete gamma functions are not stable.

For large values of x and m, the terms $x^m e^{-x}$ and $(m+1/2)!\ \text{erfc}(x)$ occurring in Eq. (21) can easily lose precision. To obtain benchmark results, we employ 100 digit arithmetic as afforded by the MPFUN suite of multi-precision programs [37], and the identity

$$\mathcal{G}_m(\beta,r) = 2\beta^{2m+1} r^m F_m(\beta^2 r^2) , \tag{23}$$

where the associated incomplete gamma functions [38–41]

$$F_m(t) = \frac{\gamma(m+1/2,t)}{2t^{m+1/2}} \tag{24}$$

are computed with the downward recurrence

$$F_m(t) = \frac{1}{2m+1}\left(2tF_{m+1}(t) + e^{-t}\right) \tag{25}$$

when $t < 300$, and starting from $m = 1200$. For $t \geq 300$, the approximation

$$F_m(t) = \frac{\pi^{1/2}(2m-1)!!}{2^{(m+1)}t^{(m+1/2)}} \tag{26}$$

is used. This approach guarantees very high accuracy even for large m, and will be referred to as the benchmark method. Note that there are other stable and efficient approaches for computing the incomplete gamma functions $\gamma(n,x)$ and $\Gamma(n,x)$, which involve continued fractions and series expansions [31,35,42].

In Table 1, results accurate to 32 digits are listed for $\mathcal{M}_l^m(\text{ws})$ with $l \leq 20$, ws=1 and ws=2. For small l, the benchmark and standard methods agree to within 14 digits. However, by $l = 40$, the standard method is accurate to within 10 digits, and by l=70 the standard method only retains 3-2 digits. Note that in the FMM, using poles of order 20 involves interaction tensors of order 40. Clearly, the propagation of these errors in simulation depends on the magnitude of higher order moments. In addition to reliability, accurate high order $\mathcal{M}$s are useful in the comparison of results from multipole code with those obtained "exactly" using the Ewald method, as well as in the calibration of multipole truncation errors.

B. Implementation

The FMM enables linear scaling computation of the potential and forces for a system of point charges by separating the interactions into two regimes that both attain an $\mathcal{O}(N)$ computational complexity, the near-field and the far-field. The near-field employs conventional direct methods, while far-field interactions are computed with a hierarchy of multipole expansions. In our periodic version of the FMM, the far-field includes contributions from the lattice via the effective interaction tensor $\mathcal{M}$. Also, the near-field contributions include interactions between particles near the boundary of the unit cell and its neighboring replicas.

Contraction of the effective interaction tensor with moments of the simulation cell yields a local expansion, $S_m^l = \sum_{k,n} \mathcal{M}_{m+n}^{l+k} O_n^k$, which is the lattice contribution to the far-field potential. In conventional FMM codes, multipole moments of the simulation cell are computed as a byproduct in the first stage of the FMM algorithm (pass 1, or the upward pass). Prior to the second stage (pass 2, or the downward pass), the local expansion S_m^l is computed, then simply transferred to children boxes in the usual way. This additional step is independent of system size, with a very modest cost that depends only on p.[1]

A second contribution to the far-field potential arises for some children boxes near the boundaries of the simulation cell. In the non-periodic FMM, boxes near the edge of the system have a reduced number of interacting neighbors due to the finite extent of the system. In the periodic case, all boxes have an identical number of neighbors, with the children boxes at the edge of the simulation cell interacting with periodically replicated boxes. Similarly, boxes that are not well-separated near these boundaries require modifications to their near-field potential. The additional near-field and far-field work due to these boundary effects is reduced, relative to the total, as the size of the system grows. The additional interactions actually simplify book-keeping since all cells interact with identical sets of neighboring boxes.

While conventional implementations of the FMM encumber $\mathcal{O}(p^4)$ cpu dependence, our periodic FMM employs coordinate rotations [43] to achieve a reduced $\mathcal{O}(p^3)$ computational complexity. Our non-periodic FMM differs by virtue of the fractional tiers method [44], which keeps the number of particles per lowest level box constant, but this type of procedure is inapplicable to periodic systems.

C. Timing

Timings for our implementation of the p^3 periodic FMM have been obtained for a cubic simulation cell filled with randomly placed charges of magnitude ± 1. As particles are added to the system, the simulation cell dimensions are increased to maintain a constant density of unity. For example, increasing N by a factor of 8 results in doubling the simulation cell width, height, and depth. For numerical stability, the particle charges and positions within the simulation cell are mapped to the unit cube. Following the FMM, the results are transformed back to the appropriate dimensions to obtain the correct values.

All timings were obtained using a 400 MHz Alpha 21164 processor, which performs at roughly 14 SpecFP95. To illustrate scalability of the algorithm, the data reported in Figures 2-5 are for $N \in [10^3, 10^6]$. In addition to particle number, we have explored the dependence of cpu time with p=6, 12, 18 and ws=1 and 2. Specifically, Figures 2 and 3 are log-log plots of the cpu time required to evaluate the potential and forces at each particle in the simulation cell, for ws=1 and ws=2 respectively. These curves establish an effective linear scaling over three orders, as measured by tangents either above or below the piecewise quadratic curves. Slopes of these tangents are unity to two significant figures, as guaranteed by formulation

FIG. 2. Periodic FMM cpu times for ws=1 with p=6, p=12, and p=18.

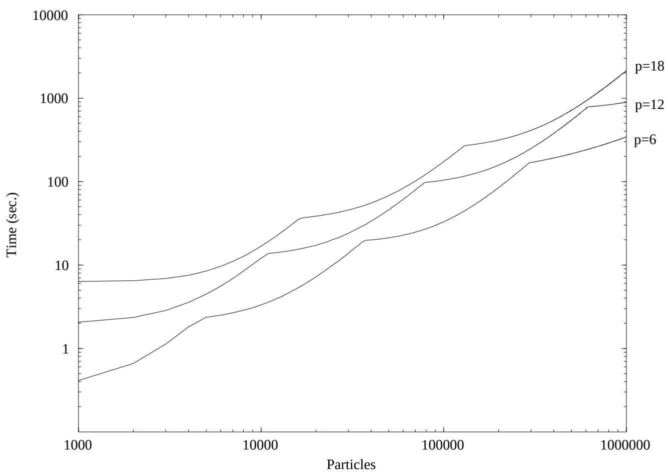
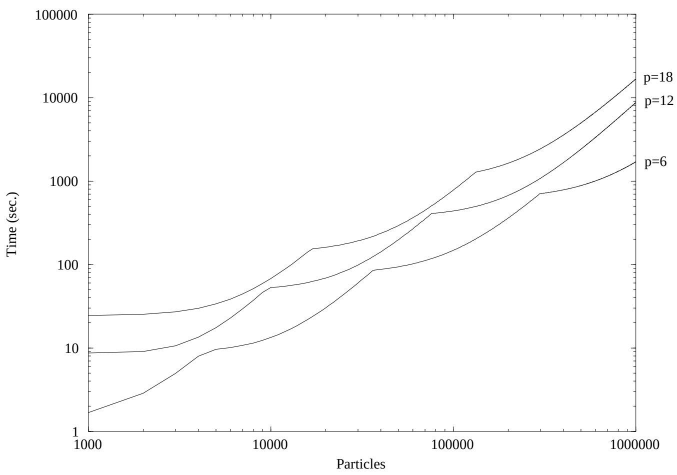


FIG. 3. Periodic FMM cpu times for ws=2 with p=6, p=12, and p=18.



of the FMM. The piecewise quadratic curves are a well-known feature of the FMM [22], with each local curve corresponding to a tree of given depth. Our non-periodic FMM differs by virtue of the fractional tiers method [44], which keeps the number of particles per leaf node constant. Unfortunately, this type of procedure is inapplicable to periodic systems.

The errors reported in Figures 4 and 5 are absolute RMS errors in the potential, computed at each point, and are a direct measure of the error that is bounded by the FMM. In Figures 4 (ws=1) and 5 (ws=2), the logarithm of the absolute RMS error in the potential is shown, corresponding to the data of Figures 2 and 3. Values of the potential computed with (p=20, ws=2) were used as the benchmark. These plots are to a good approximation linear, with a slope that is approximately 1/6 for all parameter sets. This gradual $N^{1/6}$ error accumulation is analogous to the round-off error resulting from finite precision arithmetic. Because the $N^{1/6}$ error accumulation is so gradual, a given set of parameters (p, ws) maintains an essentially constant precision over accessible particle numbers, just as 64 bit floating point arithmetic is adequate over a large range of problem sizes despite the

[1]Because computing the lattice field is so cheap, it may be worthwhile to decouple it from the FMM in order to use larger p. In this way, high accuracy lattice contributions may be achieved with only a nominal increase in cpu time.

FIG. 4. Absolute RMS errors in the potential obtained with ws=1 and p=6, p=12, and p=18.

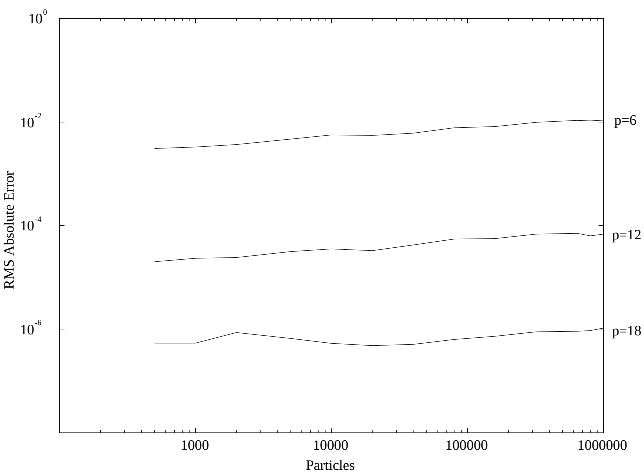
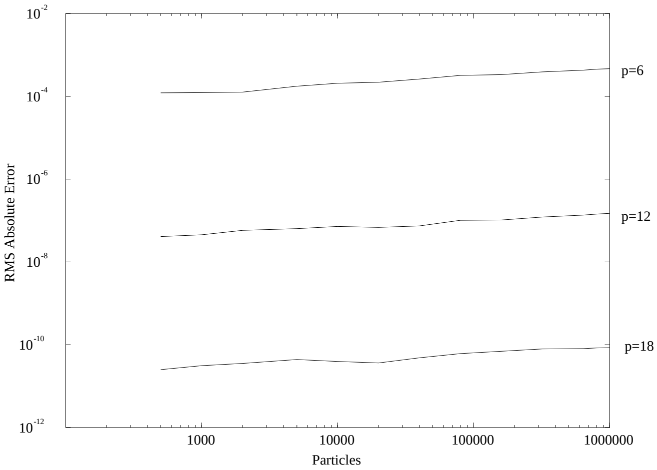


FIG. 5. Absolute RMS errors in the potential obtained with ws=2 and p=6, p=12, and p=18.

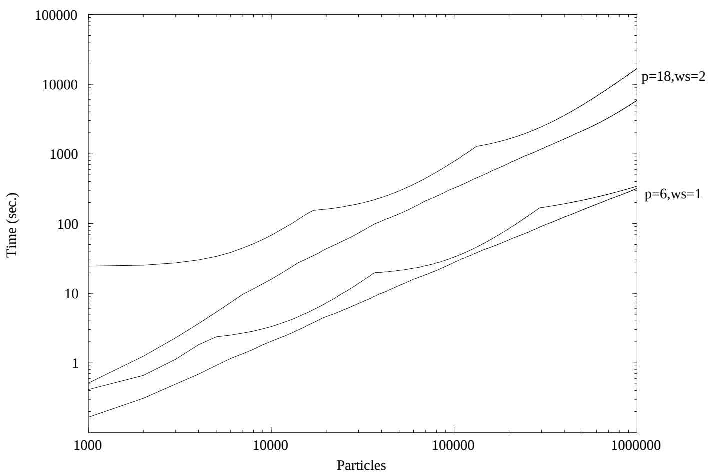


accumulation of round-off errors.

Recently, it has been argued that p should increase with system size in order to maintain a constant level of precision, and that the FMM is therefore not $\mathcal{O}(N)$ [45,46]. However, by the same logic, arithmetic should be performed in variable precision according to the size of the problem, which would require the computational complexity of *every* numerical algorithm to be reassessed! With this argument, the classical N^2 algorithm for performing Coulomb interactions is really steeper than quadratic because the precision with which the square roots and accumulations are performed must increase with system size in order to maintain the same RMS errors in the forces or potential.

Taken together, Figures 2-5 clarify the trade offs in precision and performance related to changes in ws and p. Employing ws=2 is essential for high accuracy calculations (albeit at the expense of significantly increasing the number of direct interactions for a given depth of tree). The tradeoff between ws=1 and ws=2 in terms of cost is a direct reflection of the relative efficiency of near-field and far-field interactions, and is thus implementation dependent. In our code, the high

FIG. 6. Timings for the periodic and non-periodic FMM for (p=6,ws=1) and (p=18,ws=2).



relative efficiency of far-field interactions makes ws=2 calculations slower than ws=1 calculations with higher p. Thus (p=12,ws=1) calculations are more efficient and more accurate than (p=6,ws=2).

Finally, since our periodic FMM is a direct extension of our established non-periodic FMM methods [22,44,43], it is of interest to compare their performance. Such a comparison is presented in Figure 6 for (p = 6,ws=1) and (p=18,ws=2) as extremes of efficiency and accuracy, respectively. The non-periodic curves are unscalloped due to the fractional tiers method which, as aforementioned, is not applicable in the periodic implementation. A conventional non-periodic code would yield a series of piecewise quadratic curves to which the fractional tiers curve would be a lower tangent. The small difference between this limit, and the curves shown in Figure 6, clearly illustrates the very minimal cost associated with applying periodic boundary conditions in the FMM.

D. Comparisons

As discussed in the introduction, the FMM and the fast Ewald methods are the methods of choice for treating large periodic systems. Here, we compare the performance of our new periodic FMM with other implementations and the fast Ewald methods, with an emphasis on qualitative features, as reported timings may depend critically on the computer employed, the compiler options and math libraries used, the model system chosen, *etc.*

1. Previous implementations of the periodic FMM

a. Schmidt and Lee The early implementation of Schmidt and Lee [11] is an $\mathcal{O}(p^4)$ analog of our present method. We have already shown that their non-periodic implementation is substantially slower than our own [43]. Therefore, similar conclusions follow directly for the periodic method.

b. Figueirido et al. The recent work of Figueirido *et al.* [14] is an alternative p^4 FMM implementation equivalent to Schmidt and Lee, but emphasizing the use of multiple time step integrators in combination with the FMM, as pioneered by Windemuth [47]. For (p>13,ws=1,s=4) they observe identical timings for systems of 45,000 and 90,000 charges, indicating that far-field multipole manipulation is completely dominant. By contrast, going from 90,000 to 45,000 particles with (p=18,ws=1,s=4) we observe roughly a factor of two decrease in computer time, indicating the efficiency improvements attained by our p^3 method, for high p.

c. Pollock and Glosli The FMM implementation of Pollock and Glosli [10] is based on an FFT treatment of the translation operators, which reduces the computational dependence on p to $p^2 \log p$ at the cost of substantial implimentational complexity and memory overhead [48]. In their implimentation, it is not clear exactly how periodic boundary conditions are imposed. They report p=7 FMM timings for relatively small systems (≤20,000 particles). For N = 20K, and (p=7,ws=1,s=3), their method appears to be about twice as fast as our p^3 method for the far-field component. However, our p^3 method is simpler to implement, scales readily to very large N, and admits the use of deep trees without inccuring an excessive memory overhead.

2. Periodic FMM vs fast Ewald

a. Pollock and Glosli For small systems (<20,000 particles) and relatively low accuracy (p=7, ws=1), Pollock and Glosli [10] have compared FMM against PPPM, and conclude that the latter is approximately four times faster. As our method appears slightly slower than theirs for these small system sizes (see paragraph c above), we can only reinforce this conclusion.

b. Essman et al. For roughly similar conditions, Figueirido *et al.* [14] conclude that their periodic FMM is roughly three times slower than the smooth particle mesh Ewald (SPME) method of Essman *et al.* [9], which is comparable to PPPM in computational effort [49]. Since we estimate that our method is roughly a factor of 6-8 slower than PPPM, but well ahead of Figuerido *et al.*, we can only conclude that quantitative comparisons of this sort should be interpreted with caution.

c. Time as a function of accuracy The computational cost of the periodic FMM appears to increase much less steeply than the fast Ewald methods. For example, Essman *et al.* [9] report data showing that roughly an order of magnitude more computer time is required to increase precision by four digits. As the periodic FMM curves are piecewise quadratic, the corresponding ratio is size dependent, but inspection of Figs 2-4 suggests that an increase in precision of 6 to 8 digits is achieved with an order more computational effort.

d. N vs $N \log N$ From a formal point of view, the periodic FMM is a true $\mathcal{O}(N)$ method, while the fast Ewald methods, based on the 3D FFT, are $\mathcal{O}(N\log N)$. This suggests that for sufficiently large systems the periodic FMM will ultimately be preferred, but the position of this crossover remains, in our opinion, an outstanding question. Pollock and Glosli put this crossover at $N = 10^{60}$, based on extrapolation from N=1, 5, 10, and 20K. As suggested by Figures 2 and 3, this estimate may be the result of limited sampling. At the other extreme, Toukmaji and Board [5] put the cross over at about $N = 10^5$.

IV. CONCLUSIONS

In summary, attention has been drawn to the seminal work of Nijboer and De Wette [19–21], and their central result has been recast in a simplified form suitable for modern multipole algorithms that employ the solid harmonics. The distinction between the extrinsic values obtained by lattice sums of the multipole interaction tensor, and the intrinsic values that are associated with Taylor's expansion of the Ewald formulae, has been made. Also, the universal constants $\mathcal{M}$ for the unit cube with ws=1 and ws=2 have been computed to 32 digits using extended precision arithmetic.

We have presented results spanning three orders of magnitude in system size, and roughly eight orders of magnitude in RMS deviation in the potential, to indicate the performance of our implementation. Qualitative comparisons against previous periodic FMM methods and alternative particle-particle particle-mesh Ewald methods are attempted. In general we confirm previous conclusions that fast Ewald methods are preferable for systems in the range of 10^4 particles when only moderate accuracy is desired. For larger particle numbers, the situation is less clear cut and we caution against quantitative conclusions drawn from limited extrapolations and cross-platform comparisons. It is clear however, that relative to the fast Ewald methods, the periodic FMM becomes increasingly competitive with increasing accuracy and particle number. This is of particular interest in the context of electronic structure calculations, which is a problem on which we will report shortly [50].

Finally in comparing against the fast Ewald methods it is important to recognize that while we believe our FMM implementation is state-of-the-art, it is not the end of the line. FMM techniques are still undergoing very rapid development and all such developments will translate directly to improvements in the periodic FMM framework presented here. Furthermore the ability to use *any* fast method for the evaluation of the central interactions means that the formulation presented here may be useful for adapting a wide range of fast methods [34] to treat periodic systems.

V. ACKNOWLEDGEMENTS

This work was supported by the Laboratory Directed Research and Development Program of the Ernest Orlando Lawrence Berkeley National Laboratory under the U.S. Department of Energy, Contract No. DE-AC03-76SF00098. Computational resources were provided by the University of Minnesota-IBM Shared Research Project.

APPENDIX A

Following Nijboer and De Wette (NDW) [19], the smooth component of the effective interaction tensor is evaluated with Parseval's theorem. The similarity of the NDW approach to the earlier work of Kanamori *et al.* [51] should be pointed out. The present work is different from that of NDW in that only a lattice of unit cells is considered to simplify the notation, intermediate steps have (hopefully) been clarified, and the irregular solid harmonic interaction tensor is considered, rather than the conventional one.

Ignoring for the moment the limits on $\mathbf{n}$, the smooth component of the effective interaction tensor, given by Eq. (16), may be expressed as

$$B_l^m(\beta) = \sum_{\mathbf{n}} v_l^m(\beta, \mathbf{n}) \tag{27}$$

where

$$v_l^m(\beta, \mathbf{r}) = \tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{r}}) e^{im\phi_{\mathbf{r}}} \mathcal{F}_l(\beta, r)\ . \tag{28}$$

A rapidly convergent expression for Eq. (27) will be derived, and the neighboring boxes subtracted off explicitly to arrive at Eq. (17). To obtain this reciprocal space expression, Parseval's theorem [52,53]

$$\begin{aligned} B_l^m(\beta) &= \int \mathrm{d}\mathbf{r}\ w(\mathbf{r})\ v_l^m(\beta, \mathbf{r}) \\ &= \int \mathrm{d}\mathbf{k}\ W(\mathbf{k})\ V_l^m(\beta, \mathbf{k}) \end{aligned} \tag{29}$$

is invoked, where W and V are the Fourier transforms of w and v. In Eq. (29),

$$w(\mathbf{r}) = \sum_{\mathbf{n}} \delta(\mathbf{n} - \mathbf{r}) \tag{30}$$

is the analytic function that relates v_l^m to B_l^m by convolution. In this instance, application of Parseval's theorem is equivalent to direct spherical summation of Eq. (27) [19]. All that remains is to find the Fourier transforms W and V_l^m

The Fourier transform of w may be obtained simply with the following steps:

$$\begin{aligned} W(\mathbf{k}) &= \frac{1}{(2\pi)^{3/2}} \int \mathrm{d}\mathbf{r}\ e^{i\mathbf{k}\cdot\mathbf{r}} \sum_{\mathbf{n}} \delta(\mathbf{n} - \mathbf{r}) \\ &= \frac{1}{(2\pi)^{3/2}} \sum_{\mathbf{n}} e^{i\mathbf{k}\cdot\mathbf{n}} \\ &= (2\pi)^{3/2} \sum_{\mathbf{n}} \delta(\mathbf{k} - 2\pi\mathbf{n})\ . \end{aligned} \tag{31}$$

On going from the second to the last line in Eq. (31), the identity Eq. (31) of Ref. [53] on P. 1437 was employed.

Obtaining a closed form for the Fourier transform of v_l^m is more involved, and relies on the expansion [54]

$$\begin{aligned} e^{i\mathbf{k}\cdot\mathbf{r}} = &\frac{(2\pi)^{3/2}}{4\pi\sqrt{kr}} \sum_{a=0}^{\infty} i^a J_{a+1/2}(kr) \\ &\times \sum_{b=-a}^{a} \frac{(b-a)!}{(b+a)!} P_a^b(\cos\theta_{\mathbf{r}}) e^{-ia\phi_{\mathbf{r}}} P_a^b(\cos\theta_{\mathbf{k}}) e^{ia\phi_{\mathbf{k}}}\ . \end{aligned} \tag{32}$$

Switching to spherical polar coordinates, and employing the identities

$$\int_0^\pi P_m^l(\cos\theta) P_a^b(\cos\theta)\ \sin\theta \mathrm{d}\theta = \frac{2}{2l+1} \frac{(l+m)!}{(l-m)!} \delta_{l,a} \tag{33}$$

and

$$\int_0^{2\pi} e^{i\phi(m-b)} \mathrm{d}\phi = 2\pi\delta_{m,b}\ , \tag{34}$$

the intermediate result

$$\begin{aligned} V_l^m(\beta, \mathbf{k}) &= \frac{1}{(2\pi)^{3/2}} \int \mathrm{d}\mathbf{r}\ e^{i\mathbf{k}\cdot\mathbf{r}} v_l^m(\beta, \mathbf{r}) \\ &= \frac{i^l}{\sqrt{k}}\ \tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{k}}) e^{im\phi_{\mathbf{k}}} \\ &\qquad \times \int_0^\infty \mathrm{d}r\ r^{3/2} J_{a+1/2}(kr) \mathcal{F}_l(\beta, r) \end{aligned} \tag{35}$$

obtains. Using the identity

$$\mathcal{F}_l(\beta, r) = \frac{2r^l}{\Gamma\left(l + \frac{1}{2}\right)} \int_0^\beta u^{2l} e^{-u^2 r^2} \mathrm{d}u \tag{36}$$

and switching the order of integration in the last line of Eq. (35), we find (see Appendix A.4 of Ref. [52])

$$\begin{aligned} \int_0^\beta \mathrm{d}u\ u^{2l} \int_0^\infty \mathrm{d}r\ r^{\frac{3}{2}+l} e^{-u^2 r^2} J_{l+\frac{1}{2}}(kr)\, e^{-u^2 r^2} \\ = 2^{1/2-l} k^{l-3/2} e^{-k^2/4\beta^2}\ . \end{aligned} \tag{37}$$

Substituting this result into Eq. (35) yields

$$V_l^m(\beta, \mathbf{k}) = \frac{i^l k^{l-2} e^{-k^2/4\beta^2}}{2^{l-1/2}\Gamma\left(l + \frac{1}{2}\right)}\ \tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{k}}) e^{im\phi_{\mathbf{k}}}\ . \tag{38}$$

Finally, combining Eqs. (29), (31), and (38), the rapidly convergent reciprocal space sum

$$\begin{aligned} \mathcal{B}_l^m(\beta) = &\frac{i^l \pi^{l-1/2}}{\Gamma(l+1/2)} \\ &\times \sum_{\mathbf{n}} n^{l-2} e^{-n^2\pi^2/\beta^2}\ \tilde{\tilde{P}}_l^m(\cos\theta_{\mathbf{n}}) e^{im\phi_{\mathbf{n}}} \end{aligned} \tag{39}$$

is obtained. To make the corresponding multipole expansion converge rapidly, contributions from the neighboring cells must be subtracted, which leads directly to Eq. (17).

[1] A. Redlack and J. Grindlay, Can. J. Phys. **50**, 2815 (1972).
[2] S. Stuart, J. Comp. Phys. **29**, 127 (1978).
[3] S. W. De Leeuw, J. W. Perram, and E. R. Smith, Proc. R. Soc. Lond. A **373**, 27 (1980).
[4] M. Allen and D. Tildesley, *Computer Simulation of Liquids* (Oxford Science, London, 1990).

[5] A. Toukmaji and J. A. Board Jr., Comp. Phys. Comm. **95**, 73 (1996).
[6] J. E. Roberts and J. Schnitker, J. Chem. Phys. **101**, 5024 (1994).
[7] R. Hockney and J. Eastwood, *Computer simulation using particles* (McGraw-Hill, New York, 1981).
[8] B. Luty, M. Davis, I. Tironi, and W. van Gunsteren, Mol. Simm. **14**, 11 (1994).
[9] U. Essmann *et al.*, J. Chem. Phys. **103**, 93 (1995).
[10] E. L. Pollock and J. Glosli, Comp. Phys. Comm. **95**, 93 (1996).
[11] K. E. Schmidt and M. A. Lee, J. Stat. Phys. **63**, 1223 (1991).
[12] J. K. Singer, EW. J. Numer. Math. **3**, 199 (1995).
[13] C. G. Lambert, T. A. Darden, and J. A. Board, J. Comp. Phys. **126**, 274 (1996).
[14] F. Figueirido, R. M. Levy, R. Zhou, and B. Berne, J. Chem. Phys. **106**, 9835 (1997).
[15] K. Hinsen, Technical Report No. AMER.92.015, Shell Research, Amsterdam (unpublished).
[16] H. Ding, N. Karasawa, and W. A. Goddard III, Chem. Phys. Let. **196**, 6 (1992).
[17] J. Shimada, H. Kaneko, and T. Takada, J. Comp. Chem. **14**, 867 (1993).
[18] J. Shimada, H. Kaneko, and T. Takada, J. Comp. Chem. **15**, 28 (1994).
[19] B. R. A. Nijboer and F. W. De Wette, Physica **23**, 309 (1957).
[20] B. R. A. Nijboer and F. W. De Wette, Physica **24**, 422 (1958).
[21] F. W. De Wette and B. R. A. Nijboer, Physica **24**, 1105 (1958).
[22] C. A. White and M. Head-Gordon, J. Chem. Phys. **101**, 6593 (1994).
[23] H. Y. Wang and R. Le Sar, J. Chem. Phys. **104**, 4173 (1996).
[24] J. M. Pérez-Jordá and W. Yang, J. Chem. Phys. **104**, 8003 (1996).
[25] M. Challacombe and E. Schwegler, J. Chem. Phys. **106**, 5526 (1997).
[26] M. Challacombe, E. Schwegler, and J. Almlöf, Chem. Phys. Lett. **241**, 67 (1995).
[27] M. Challacombe and E. Schwegler (unpublished).
[28] B. Delley, J. Phys. Chem. **100**, 6107 (1996).
[29] K. Hinsen and B. U. Felderhof, J. Math. Phys. **33**, 3731 (1992).
[30] D. J. Adams and G. S. Dubey, J. Comp. Phys. **72**, 156 (1987).
[31] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipies in Fortran* (Cambridge University Press, Port Chester, New York, 1992).
[32] B. C. Carlson and G. S. Rushbrooke, Proc. Cambridge Philos. Soc. **46**, 626 (1950).
[33] C. Berman and L. Greengard, J. Math. Phys. **35**, 6036 (1994).
[34] A. Graps, Amara's Recap of Particle Simulation Methods, this excellent overview of the N-body family tree may be found at http://www.amara.com/papers/nbody.html.
[35] *Handbook of Mathematical Functions*, ninth ed., edited by M. Abramowitz and I. A. Stegun (Dover Publications Inc., New York, 1987).
[36] M. H. Cohen and F. Keffer, Phys. Rev. **99**, 1128 (1955).
[37] D. H. Bailey, ACM Trans. Math. Software **19**, 288 (1993).
[38] I. Shavitt, Meth. Comp. Phys. **2**, 1 (1963).
[39] L. L. Shipman and R. E. Christoffersen, Comp. Phys. Comm. **2**, 201 (1971).
[40] L. J. Schaad and G. O. Morrell, J. Chem. Phys. **54**, 1965 (1971).
[41] F. E. Harris, Int. J. Quant. Chem. **23**, 1469 (1983).
[42] A. R. Didonato and A. H. Morris, ACM Trans.on Math. Software **12**, 377 (1986).
[43] C. A. White and M. Head-Gordon, J. Chem. Phys. **105**, 5061 (1996).
[44] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **257**, 647 (1996).
[45] S. Aluru, G. M. Prabhu, and J. Gustafson, in *Proceedings SUPERCOMPUTING '94* (IEEE Computer Society Press, Los Alamitos, CA, 1995), pp. 420–6.
[46] S. Aluru, J. Sci. Comp. **17**, 773 (1996).
[47] A. Windemuth, in *Parallel Computing in Computational Chemistry* (ACS Books, Washington, DC, 1995).
[48] W. D. Elliott and J. A. Board Jr., Siam J. Sci. Comp. **17**, (1996).
[49] T. Darden, personal communication.
[50] Work in progress.
[51] J. Kanamori, T. Moriya, K. Motizuki, and T. Nagamiya, J. Phys. Soc. Japan **10**, 93 (1955).
[52] I. N. Sneddon, *Fourier Transforms* (McGraw-Hill, New York, 1951).
[53] C. Cohen-Tannoudji, B. Diu, and F. Laloë, *Quantum Mechanics* (Wiley and Sons, New York, 1977), Vol. II.
[54] See Ref. [51] and also Ref. [55] P. 497, Eq. (B.105), or Ref. [53] P.948, Eq. (65). Note that $j_l(x) = \sqrt{\pi/2x}J_{l+1/2}(x)$.
[55] A. Messiah, *Quantum Mechanics* (John Wiley and Sons, New York, 1958), Vol. I.

TABLE I. The effective interaction tensor elements for a spherically summed lattice of unit cells through order $l = 20$.

l	m	$\mathcal{M}_l^m(\mathrm{ws}=1)$	$\mathcal{M}_l^m(\mathrm{ws}=2)$
4	0	2.8119304871888668206200481879919D+00	9.8485467452574389829712692036430D-01
4	4	1.4059652435944334103100240939959D+01	4.9242733726287194914856346018220D+00
6	0	5.4795908739321644069327702900830D-01	6.5623422860067524157994035757200D-02
6	4	-3.8357136117525150848529392030580D+00	-4.5936396002047266910595825030040D-01
8	0	1.2156157302097918942115948482762D+02	-6.8626119152683450863254261961150D-01
8	4	1.2156157302097918942115948482762D+02	-6.8626119152683450863254261961150D-01
8	8	7.9015022463636473123753665137950D+03	-4.4606977449244243061115270274740D+01
10	0	3.1179916736109123107822587274280D+02	-1.2457280551463641551202434559869D-01
10	4	-6.8595816819440070837209692003420D+02	2.7406017213220011412645356031711D-01
10	8	-1.1661288859304812042325647640581D+04	4.6590229262474019401497105253910D+00
12	0	2.4245612747359092217199640516493D+05	4.1692731501913177016791478604950D+02
12	4	2.0375858264140266510162557633886D+05	4.1061218410741854245075379713950D+02
12	8	7.0682666545985000701644635107790D+05	4.9270888595969050277384665296890D+02
12	12	2.3702435984527078287639617913271D+08	4.2547490401383123579011584199450D+05
14	0	2.0954087119885542648713979286402D+06	2.8616631637228685773233790308573D+02
14	4	-2.6940969154138554834060830511089D+06	-3.6792812105008310279872016111020D+02
14	8	-1.7062613797621084728238525990356D+07	-2.3302114333171929843918943536981D+03
14	12	-6.5406686224214158124914349629700D+08	-8.9324771610492397735022616891760D+04
16	0	5.4279858299650169624382885076980D+08	6.3709709719614919905762046729900D+05
16	4	2.2841041529105412872383486949334D+08	5.9486986304961071015286445862700D+05
16	8	1.2973301854895758582918144058332D+09	7.3844245914784157242903488811180D+05
16	12	2.5882484900055575638355343741650D+10	4.0406121694071514047809547662630D+06
16	16	6.9973653547984205656745320248030D+12	1.0184121261186984461485623112385D+10
18	0	1.4686049951258450810632984509870D+10	1.7833814404746090864126676944445D+06
18	4	-1.5376346487994712576061405817891D+10	-1.9097270960159450077253196645607D+06
18	8	-2.4226921558569614141580552133902D+10	-2.2373412064676267995353888960110D+06
18	12	-1.0692416604738659056152567751127D+12	-1.3897171037814018266686814699226D+08
18	16	-3.9585194668444324327226917843820D+13	-4.6533202641465144258077188159160D+09
20	0	2.9414124910043233182340700935067D+12	1.1053789653219762994427196549037D+09
20	4	5.0799363324581667612792095666680D+11	8.2755105319168490831988163978850D+08
20	8	4.3319375525806128280090124574150D+12	1.2641377722535713800843413778266D+09
20	12	4.7785845726839660008475961329560D+13	6.2253504888659176501350202191680D+09
20	16	1.6103883836731949966180674427718D+15	1.8463055977824588952115743135381D+11
20	20	5.0103139602723200237451484155740D+17	2.5893801321268771887525639926163D+14