Journal of Statistical Physics, Vol. 63, Nos. 5/6, 1991

Implementing the Fast Multipole Method in Three Dimensions

K. E. Schmidt[1] and Michael A. Lee[2]

Received January 29, 1991

The Rokhlin–Greengard fast multipole algorithm for evaluating Coulomb and multipole potentials has been implemented and analyzed in three dimensions. The implementation is presented for bounded charged systems and systems with periodic boundary conditions. The results include timings and error characterizations.

KEY WORDS: Fast multipole method; many-body problem; N-body problem.

1. INTRODUCTION

Prior to the dissertation work of Greengard,[1] the statistical mechanics of Coulomb systems by simulation methods utilized algorithms which required order N^2 or $N^{3/2}$ operations to calculate the potential energy of N charges or multipoles. The N^2 algorithm is simply the definition of the potential energy of a set of charges q_i, located at positions s_i, $1 \leqslant i \leqslant N$,

$$\mathrm{PE} = \sum_{i<j}^{N} \frac{q_i q_j}{s_{ij}}$$

In systems with periodic boundary conditions, the standard Ewald sum method[7] requires order $N^{3/2}$ operations. The $N^{3/2}$ and some approximate order N methods have been reviewed by Greengard[2] and others[3,4] and will not be discussed here.

The work of Greengard and Rokhlin[5] described the theoretical foundations of an order N algorithm in two and three dimensions which they termed the fast multipole method. The fast multipole method in two

[1] Department of Physics, Arizona State University, Tempe, Arizona 85287-1504.
[2] Department of Physics, Kent State University, Kent, Ohio 44242.

0022-4715/91/0600-1223$06.50/0

dimensions is superior to conventional schemes for as few as 100 particles, as shown by explicit numerical tests.[1,6]

While development of the 3D implementation has begun on several fronts,[7,8] detailed numerical tests of the Rokhlin–Greengard algorithm have not been forthcoming. Since the foundations of the method are simple and elegant, one can only conjecture that its implementation has been delayed because naive estimates of its breakeven point can easily be beyond capabilities of current machines.

The purpose of this report is to characterize the efficiency and accuracy of the order N algorithm in its original formulation.[1] This is an algorithm of enormous implication in computational physics. Therefore, we will describe the computational implementation and difficulties and its usefulness relative to current hardware. Finally, we provide timings and empirical tests of its accuracy. The reader who is interested in the analysis of algorithmic complexity and mathematical criteria for precision are referred to the original articles.[1,8]

2. THE ORDER-N ALGORITHM

We give a descrition, at a practical level, of the fast multipole algorithm and restate the relevant theorems using familiar notation. The mathematical conventions for spherical harmonics and units are those found in Jackson.[9]

The economy of the fast multipole method is based on the representation of interactions between distant charged regions using truncated multipole expansions. The algorithm begins by logically dividing the charged system into approximately N cubic subdomains. The multipole expansions of the potential due to charges in each subdomain are combined in a hierarchical procedure eliminating redundant operations to obtain an external potential for each box. The potential of the individual particles is then expressed as a direct interaction with particles in nearby boxes plus a Taylor series expansion of the field due to particles in all distant boxes. The combining of the multipole expansions is done such that only order N operations are required and a fixed precision is maintained.

The level of precision ε, which is a rigorous bound, is guaranteed to be met when all multipole are carried out to an order $L = -\log_2(\varepsilon)$. This bound is generous in physical applications. The proofs of the bounds needed are given by Greengard[1] and will not be repeated here.

3. MULTIPOLE TRANSFORMATIONS

The fast multipole method requires three transformations of the expansions of the potential. The transformations begin with a multipole expansion of the potential,

$$\Phi(\mathbf{r}) = 4\pi \sum_{l,m} \frac{M_{lm} Y_{lm}(\theta, \Phi)}{(2l+1) r^{l+1}} \tag{1}$$

or a local expansion

$$\Phi(\mathbf{r}) = 4\pi \sum_{l,m} L_{lm} r^l Y_{lm}(\theta, \Phi) \tag{2}$$

that is given in spherical coordinates (r, θ, Φ) relative to an origin. The multipole moments are defined by Jackson,[9]

$$M_{lm} = \sum_i q_i s_i^l Y_{lm}^*(\theta_i, \Phi_i) \tag{3}$$

and the L_{lm} we will refer to as the local moments of the Taylor series expansion.

We define the coordinate systems at origin O and O' as in Fig. 1 with $\mathbf{r}_t$, designating the translation from O to O'. Given L_{lm} or M_{lm} of a potential expansion in terms of $\mathbf{r}$ relative to O, the following three transformations

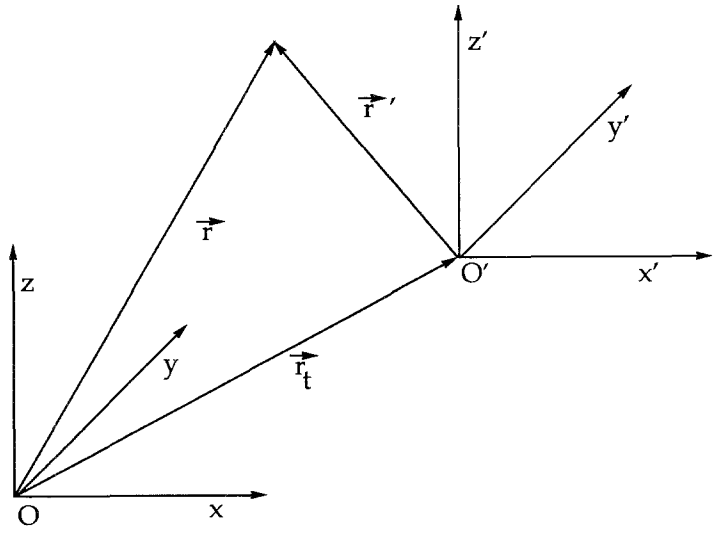

Fig. 1. Coordinate systems employed in multipole transformations.

are needed by the fast multipole method to obtain the moments $L'_{l'm'}$ or $M'_{l'm'}$ of a potential expansion in terms of $\mathbf{r}'$ relative to O':

$$M'_{l'm'} = \sum_{l,m} T^{MM}_{l'm',lm} M_{lm} \tag{4}$$

$$L'_{l'm'} = \sum_{l,m} T^{LM}_{l'm',lm} M_{lm} \tag{5}$$

$$L'_{l'm'} = \sum_{l,m} T^{LL}_{l'm',lm} L_{lm} \tag{6}$$

Defining

$$a_{\mathrm{lm}} = (-1)^{l+m} \frac{(2l+1)^{1/2}}{[4n(l+m)!\,(l-m)!]^{1/2}} \tag{7}$$

we have the following formulas for the transformation matrices:

$$T^{MM}_{l'm',lm} = 4\pi \frac{(-r_t)^{l'-l}\, Y^*_{l'-l,m'-m}(\theta_t, \Phi_t) a_{l'-l,m'-m} a_{lm} (2l'+1)}{(2l+1)[2(l'-l)+1] a_{l'm'}} \tag{8}$$

$$T^{LM}_{l'm',lm} = 4\pi \frac{(-1)^{l+m}\, Y^*_{l'+l,m'-m}(\theta_t, \Phi_t) a_{lm} a_{l'm'}}{r_t^{l'+l+1}(2l+1)(2l'+1) a_{l'+l,m'-m}} \tag{9}$$

$$T^{LL}_{l'm',lm} = 4\pi \frac{r_t^{l-l'}\, Y_{l-l',m-m'}(\theta_t, \Phi_t) a_{l'm'} a_{l-l',m-m'}}{a_{lm}(2l'+1)[2(l-l')+1]} \tag{10}$$

The transformation matrices are not dense, being zero when the subscripts on the spherical harmonics are outside of their legal range.

The above transformations are exact. The multipole expansions converge outside of a sphere with radius equal to the distance to the furthest charge and the local expansion converge inside of a sphere with radius less than the distance to the nearest charge.

The fast multipole method, as described by Rokhlin and Greengard,[13] truncates all expansions and transformations to include only those spherical harmonics with l-values up to some specified maximum L. A critical feature of the implementation is maintaining a rigorous error bound. The error bound is $E = (\sum_i |q_i|)2^{-L}$ in the domain of validity of each expansion. For the truncated multipole expansion of a set of charges with M_{lm} given by Eq. 3, the bound is guaranteed for $r > 2b$, where b is the distance from the origin to the furthest charge. This domain where the rigorous bound applies we will term the domain of validity. The error bound divided by the sum of absolute charges defines a quantity $\varepsilon = 2^{-L}$ which they term the relative precision. It is maintained by the results of all transformations in

their domain of validity. Greengard[1] demonstrates that the domain of validity of a translated multipole expansion as given by the truncation of Eq. (4) is outside a sphere of radius in $r_t + a$, where r_t is the distance translated and a is the radius of the domain of validity of the original multipole expansion. Similarly, the local expansion obtained from a multipole expansion using the truncation of Eq. (5) has a domain of validity inside a sphere of radius $r_t - a$. The local-to-local transformation is exact and does not alter the domain of validity of a local expansion.

In the following description on the algorithm of Rokhlin and Greengard, the external potential for each box is obtained by combining multipole potentials in such a way that there are no redundant transformations and the potentials are used only in their domain of validity.

3.1. The Order-*N* Algorithm

We will first present the algorithm for $\acute{N}$ charges confined to a cube of sides d centered at the origin. Later, we will incorporate the changes necessary to describe a bulk system which consists of the cube and its periodic images.

Step 0. The algorithm begins by dividing the cube into successively smaller cubic subvolumes. We define a refinement level r to be the division of the cube into nonoverlapping subvolumes of dimension $d/2^r$, up to some maximum refinement R. At level $r = 0$, the refinement contains one cube, the original volume. At level $r = 1$, there are eight subvolumes, continuing to the final refinement with 8^R equal-sized subvolumes. The maximum refinement is chosen so that N is approximately 8^R. The maximum of the subsequent multipole expansion L is chosen so that $2^{-L} \leqslant \varepsilon$, the predetermined relative precision.

Step 1. Beginning at the refinement level R, calculate the 8^R truncated multipole expansions for the particles contained in each box in level R about the center of each box. This requires order L^2 operations for each box, yielding a net order NL^2.

We must now define three terms. Each box at level $r < R$ contains the eight subdivisions that make up the next greater refinement level. These eight subboxes are called child boxes. The larger box containing the child is the parent box. The term "touching box" refers to any box that shares a face, edge, or corner of a given box. Each box has 26 "touching" boxes.

Step 2. Beginning with refinement level $R - 1$ and continuing to $r = 0$, form the truncated multipole moment expansions for all charges in all boxes about their centers by using Eq. 4, to transform the multipole expansions

of the eight child boxes to the center of their parent. The translation of a multipole expansion requires L^4 operations, and this must be performed for

$$\sum_{r=0}^{R-1} 8^r$$

child boxes, the net work being order NL^4.

Step 3. Each multipole expansion will be converted to a local expansion about the center of all boxes at the same level which are sufficiently distant that the relative precision of the local expansion is ε. The multipole expansions which contribute to a local expansion with a domain of validity including the one box are the multipole expansions of all boxes at the same level which do not touch the one box. Carried out directly, this would require order N^2 work. To avoid this, a combination of local-to-local and multipole-to-local transformations is substituted.

At levels $r=0$ and 1, no boxes exist that are sufficiently distant that a valid multipole-to-local transformation can be performed. The local expansions at these levels are set to zero. Thus, beginning at level $r=2$ and continuing to $r=R$, the following operations are performed.

(a) For each box in level r, transform the local expansion of the parent box to the center of the current box. This requires L^4 operations per box.

(b) Add to the local potentials from step 3a the transformation of the multipole potentials for all boxes of the current level that satisfy the following: (i) the box does not touch the current box, and (ii) the charges in the box did not contribute to the local potential initialized in step 3a.

Step 3a requires

$$L^4 \sum_{r=2}^{R} 8^r$$

operations and step 3b requires

$$L^4 \cdot 189 \sum_{r=2}^{R} 8^r$$

operations, because there are 189 nearby boxes which have multipole expansions that can be converted to valid local expansions.

Step 4. Step 3 results in a local potential in each box hat is the potential due to all particles in all boxes at the same level which do not touch the current box.

The last step involves the calculation of the final potential and forces of all particles in the boxes at the finest level of refinement in two steps.

(a) For each box in level R, evaluate the potential and force from the local expansion at each particle location. This requires N evaluations of L^2 operations.

(b) Evaluate the remaining interaction of each particle with the other particles in the current box plus the particles in touching boxes. This operation is order N times the average number of particles in the current and neighbor boxes.

3.2. Analysis

First, one must establish that this is indeed an order-N algorithm. The choice of the number of boxes at the finest level R being approximately equal to N is critical, as in the requirement that the charge density be approximately uniform. This can be seen if we consider R fixed and not necessarily chosen to be such that $8^R \approx N$. With R fixed and N allowed to be arbitrarily larger than 8^R, we will analyze each of the above steps again.

Step 0 is an operation done only once and adds a constant to the time required and can thus be ignored.

In step 1, one initializes 8^R multipole expansions to zero, a nontrivial contribution to the overall work, taking $8^R L^2$ operations. The calculation of N multipole expansions contributes NL^2, giving a polynomial contribution, $P_1 = a8^R L^2 + bNL^2$. For step 2, the polynomial is $P_2 = c8^R L^4$, and steps 3a and 3b contribute $P_3 = d8^R L^4 + e8^R L^4$, d and e being due to step 3a and 3b, respectively. Finally, step 4a contributes $f8^R L^2$ and step 4b contributes

$$P_4 = 8^R[g_1 + g_2(N/8^R) + g_3(N/8^R)^2]$$

The overall polynomial dependence is then

$$P = a8^R L^2 + bNL^2 + c8^R L^4 + d8^R[g_1 + g_2(N/8^R) + g_3(N/8^R)^2]$$

We see that there is an N^2 dependence which is removed by relating R to N. The N^2 dependence arises from the last step, which is a direct evaluation of interactions between particles in the same box and near-neighbor boxes at the finest level. The coefficient of the N^2 term will be negligible when there is approximately one particle per box. If $N/8^R$, the average number of particles per box, becomes large, then even for a uniform charge distribution the interaction between particles in neighboring boxes contributes in important quadratic term in the N dependence. This term

then decides when one goes to a greater maximum refinement level R as N increases. Because additional refinement levels add significantly to the computational cost of the algorithm, it is desirable to have the coefficient of the quadratic term as small as possible to delay adding refinement levels as N increases.

It should be noted that it is the average of the square of the number of particles per box rather than the square of the average which determines the computational cost in the nearby potential evaluations. For very inhomogeneous systems, it is likely that the adaptive form of the fast multipole method[1] or other hierarchical methods[4] will be more appropriate.

Our computer program structure follows the steps outlined above and the polynomial dependence of each step can be separately quantified in actual application. We defer further discussion on this point until after we have presented our results. We mention only that the N^2 dependence also becomes relevant because storage limitations are encountered. Storage grows as $8^R L^2$ with a coefficient of about unity. With a modest precision requirement of 10^{-6}, $L \approx 20$. With a gigaword of memory, R cannot exceed 7.

4. PERIODIC BOUNDARY CONDITIONS

We have also developed a version of the code with periodic boundary conditions for evaluating the potential energy of a set of charges in a cube and its infinite periodic images. The usual procedure, involving Ewald sums, is well documented in many texts.[3] The Ewald sums over all infinite periodic images of each charge can be carried out in a time proportional to $N^{3/2}$.

In the fast multipole method, the procedure is initiated in the same manner as in the finite case, calculating the multipole expansion of all boxes at all refinement levels. The level-zero expansion then contains the multipole expansion for all particles in the original cube. All of the periodic images have the same multipole expansion about their centers. The fast multipole method requires the local expansion of the potential from all periodic images except the 26 nearest neighbors of the original simulation cell. Having that, the algorithm continues in its downward pass exactly as before.

In the following, we will assume a neutral system of particles. Modifications to include a uniform background to compute systems like the one-component plasma are straightforward,[3] but will not be given here.

Since the multipoles of all the images of the simulation cell are the same, they factor out of the sum over images. The transformation matrices

for each of the reqired simulation cell images can then be added to produce one transformation matrix that, when multiplied times the multipole moments of the simulation cell, produces the coefficients of the local expansion of all those images.

To construct this transformation matrix, we simply replace the values of $r_t^{-(l+1)} Y_{l,m}(\theta_t, \Phi_t)$ in Eq. (9) by its sum over values corresponding to all the images of the simulation cell except its 26 surrounding cells. The sum over all the cells except the simulation cell itself can be calculated using the Ewald method by

$$\sum_{n,o,p\neq 0,0,0} r_{nop}^{-(1+1)} Y_{lm}(\theta_{nop}\Phi_{nop})$$

$$= \sum_{n,o,p}{}' \frac{Y_{lm}(\theta,\Phi)}{(2l+1)!!}\left[\frac{2\alpha(\alpha r_{nop})^l}{\pi^{1/2}\pi}\exp(-\alpha^2 r_{nop}^2)\right.$$

$$\left. + \frac{4}{\sqrt{\pi}}\frac{2^l}{r^{l+1}} I_l + (2\pi i)^l r^{l-2}\exp-\left(\frac{\pi^2 r_{nop}^2}{\alpha^2}\right)\right] \tag{11}$$

The r_{nop}, θ_{nop}, and Φ_{nop} are the spherical coordinates of the vector $rnop = d(n\hat{x} + o\hat{y} + p\hat{z})$, α is the usual Ewald parameter selected to speed convergence, and I_l is given by the recursion relation

$$I_l = \frac{2l+1}{2} I_{l-1} + \frac{1}{2}\exp(-\alpha^2 r^2)(\alpha r)^{2l+1}$$

with

$$I_0 = \frac{\pi^{1/2}}{4}\operatorname{erfc}(\alpha r) + \frac{\alpha r}{2}\exp(-\alpha^2 r^2)$$

The contribution of the 26 surrounding cells is then subtracted explicitly. The prime on the sum indicates omission of the $n = o = p = 0$ term.

The procedure we use is equivalent to surrounding a finite set of images by a vacuum with dielectric constant $\varepsilon = 1$, and taking the limit of the number of images going to infinity. The usual Ewald method is equivalent to using a surrounding conductor with $\varepsilon = \infty$. The procedure to change from one dielectric constant at infinity to another is well known,[3] and is given by

$$V(\varepsilon) = V(\varepsilon_0) + \frac{2\pi}{V}\left(\frac{1}{3\varepsilon+1} - \frac{1}{2\varepsilon_0+1}\right)\mathbf{D}\cdot\mathbf{D} \tag{12}$$

The dipole moment,

$$\mathbf{D} = \sum_i q_i \mathbf{r}_i$$

of the simulation cube is known in terms of M_{lm} in Eq. (3).

The algorithm is modified so that step 0 creates the image multipole-to-local transformation described above, and step 3 is modified by adding the nonzero local expansion for levels $r = 0$ and $r = 1$. The local expansion for $r = 0$ in step 3a is simply given by applying this transformation to the multipole moments of the simulation cell. Step 3b does not change in its description, but in the code some logic changes. The changes are actually simplifications. In the finite system, boxes near the faces of the cube are treated differently because there are no neighboring boxes of charge outside the cube. When periodic boundary conditions are employed, all boxes, whether interior or near the surface, are treated exactly the same. For instance, at level $r = 1$, where step 3b was not performed in the finite system, one now adds to each of the local expansions of the eight subdivisions of the cube the 189 transformations of the multipole potentials of the subdivisions of the nearby periodic images. Similarly, at all further refinement levels, the full 189 transformations are required for each local expansion. This does not add significantly to the computational requirements of the algorithm, since most of the work occurs at the finest, $r = R$, level where few of the boxes are surface boxes.

5. TIMING AND ACCURACY

The above description was implemented in ASNI standard Fortran 77 on the Cray Y-MP 8/864 of the Ohio Supercomputer Center. We have carried out a sequence of tests varying the number of particles N, the refinement level R, and the order L of the expansions to obtain empirical data on timing and accuracy.

Finite Single Cube

The tests employed a distribution of charges randomly positioned in the cube and with variations in charge magnitudes randomly assigned in the interval $(-1, 1)$. Timings and precision determinations were carried out for numerous different random charge and position assignments. Table I gives the timings for a range of N, L, and R values and the mean square errors for the force and potential energy of the particles.

One can see from Table I that the breakeven point for the fast multipole algorithm depends very much on the level of refinement and number of multipoles. The error is independent of the level of refinement, as expected, and within the rigorous bounds of the theory. The optimum number of particles per box at the finest level of refinement is far more than one. For $L = 8$, it is nearly 100 at $R = 3$. This is because the N^2 term that appears with fixed R has a small coefficient. A similar set of observations can be

Table I. Comparison of Timings and Accuracies of FMM and Direct Method in a Finite Volume

N	L	R	FMM Sec	Direct Sec	Err F	Err PE
1000	8	3	13.72	0.12	0.69E-03	0.16E-03
10000	8	3	24.64	11.03	0.39E-03	0.11E-03
20000	8	3	48.25	42.34	0.26E-03	0.18E-03
40000	8	3	131.8	176.9	0.65E-04	0.12E-03
80000	8	3	448.0	677.5	0.44E-03	0.13E-03
1000	16	3	144.6	0.12	0.28E-04	0.23E-05
10000	16	3	167.0	11.03	0.16E-04	0.41E-05
20000	16	3	201.9	42.34	0.17E-04	0.31E-05
40000	16	3	310.7	176.9	0.31E-04	0.11E-04
80000	16	3	677.0	677.5	0.13E-04	0.48E-05
1000	8	4	59.97	0.12	0.18E-02	0.16E-03
10000	8	4	65.81	11.03	0.12E-02	0.27E-03
20000	8	4	73.32	42.34	0.98E-03	0.22E-03
40000	8	4	94.42	176.9	0.73E-03	0.22E-03
1000	16	4	654.7	0.12	0.48E-04	0.60E-05
10000	16	4	668.7	11.03	0.49E-04	0.17E-04
40000	16	4	736.8	176.9	0.28E-04	0.53E-05
80000	16	4	847.5	677.5	0.29E-04	0.71E-05
1000	8	5	414.3	0.12	0.10E-02	0.42E-03
10000	8	5	418.0	11.03	0.14E-02	0.18E-03
40000	8	5	437.2	176.9	0.75E-03	0.14E-03
80000	8	5	461.8	677.5	0.81E-03	0.10E-03

made for the periodic system results shown in Table II. As asserted, there is little overhead associated with treating periodic systems. We have provided timings for our Ewald calculations, but we do not offer these for comparison. Our implementation, while highly vectorized, does not employ neighbor tables and is not in other ways optimized.

Finally, we wish to add a few comments on the efficient implementation of the algorithm. For $L > 16$, $R > 3$, and $N > 10{,}000$, 98% of the CPU time in our codes involves step 3b. This is efficiently implemented in one subroutine where the innermost loop is over box indices which are the target for one of the 189 transformations. In this way, the process is highly vectorized and runs at over 180 megaflops. Because there are eight types of target boxes, each type treated separately, the possibility for a parallel implementation on an eight-processor YMP is clear. Our results and timings are all for a one-processor implementation.

In conclusion, we have shown that it is both practical and efficient to carry out the fast multipole method in three dimensions for both finite and periodic systems.

Table II. Comparison of Timings and Accuracies of FMM and Ewald Method for Periodic Boundary Conditions

N	L	R	FMM Sec	Direct Sec	Err F	Err PE
1000	8	3	13.95	8.61	0.72E-03	0.64E-03
2000	8	3	14.61	29.62	0.10E-02	0.37E-03
4000	8	3	16.77	109.3	0.44E-03	0.28E-03
10000	8	3	27.54	647.4	0.11E-02	0.71E-03
20000	8	3	59.79	2552.	0.73E-03	0.63E-03
40000	8	3	177.8	8101.	0.45E-03	0.28E-03
1000	16	3	145.0	8.61	0.90E-04	0.20E-04
2000	16	3	147.7	29.62	0.48E-04	0.25E-04
4000	16	3	151.7	109.3	0.38E-04	0.42E-04
8000	16	3	162.3	418.3	0.54E-04	0.26E-04
10000	16	3	168.2	647.4	0.27E-04	0.12E-04
1000	8	4	60.47	8.61	0.14E-02	0.60E-03
2000	8	4	60.00	29.62	0.22E-02	0.57E-03
4000	8	4	62.04	109.3	0.11E-02	0.74E-03
10000	8	4	67.37	647.4	0.58E-03	0.73E-03
20000	8	4	74.39	2552.	0.92E-03	0.72E-03
40000	8	4	99.74	8101.	0.63E-03	0.10E-02
1000	16	4	654.0	8.61	0.76E-04	0.26E-04
4000	16	4	655.9	109.3	0.59E-04	0.33E-04
8000	16	4	677.9	418.3	0.29E-04	0.24E-04
1000	8	5	415.1	8.61	0.47E-02	0.59E-03
10000	8	5	420.8	647.4	0.45E-02	0.57E-03

Our implementation is generic, following the description by Greengard.[1] Several avenues for improvement remain. First, an overall factor of two is available because the potential expansion is complex. Only half of the coefficients are actually needed. Second, Greengard and Rokhlin[7] have shown that the transformation in step 3b can be formulated as a discrete convolution and carried out using fast Fourier transforms in a tie proportional to $L \ln L$. Further possibilities exist for improvement of precision and decreasing coefficients in the polynomial dependence on CPU time. Accuracy can be improved by increasing the required precision at fixed L and decreasing the size of the domain of validity. This has the effect of increasing the number of nearby boxes that require special treatment beyond those which are touching. Greengard[1] describes such an algorithm where 875 rather than 189 nearby boxes are treated in step 3b. One may also consider modifying the requirement that all expansions and transformations are carried out to the same order L. Certainly, the most distant boxes may be more crudely represented without loss of precision. Finally, we remark that there is no theoretical requirement

that all boxes be identical cubes or even nonoverlapping. These avenues are currently being investigated by several researchers and we anticipate that an order-of-magnitude improvement should be soon forthcoming.

ACKNOWLEDGMENTS

We thank Jerry Percus for many insightful and fruitful comments and criticisms of our work over the years, and for congenial conversations over luch while both authors were at NYU. He is a man of great heart and intellect who freely shares his knowledge and friendship.

This work was carried out with an allocation of computational resources from the Ohio Supercomputer Center under grants PGS030 and PGS178. K.E.S. acknowledges support from NSF CHE90-15337. M.A.L. thanks the Department of Chemistry at New York University for their hospitality, where he was a Visiting Professor during the initial phases of this work.

REFERENCES

1. L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (MIT Press, Cambridge, 1987).
2. L. Greengard, *Comp. Phys.* **142**(Mar./Apr.) (1990).
3. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, 1990).
4. W. H. Press, in *Use of Supercomputers in Stellar Dynamics*, P. Hut and S. McMillian, eds. (Springer, 1986), p. 184.
5. L. Greengard and V. Rokhlin, *J. Comput. Phys.* **60**:187 (1985).
6. J. A. Board, Jr., R. R. Batchelor, and J. F. Leathrum, Jr., in *Proceedings of the 5th AIAA/ASME Thermophysics and Heat Transfer Conference* (1990), pp. 27–34.
7. L. Greengard and V. Rokhlin, Technical Report 602, Yale University, Department of Computer Science (1988).
8. F. Zhao, Research Report AI-TR-995, Massachusetts Institute of Technology (1987).
9. J. D. Jackson, *Classical Electrodynamics* (Wiley, New York, 1983).