An Ewald summation based multipole method

Zhong-Hui Duan[a] and Robert Krasny
Department of Mathematics, University of Michigan, Ann Arbor, Michigan 48109-1109

(Received 3 April 2000; accepted 5 July 2000)

We present a method for evaluating Coulomb interactions in periodic molecular systems. The real space term in Ewald summation is accelerated using a tree code in which interactions between clusters and distant particles are approximated by multipole expansions. The performance is reported for water systems. © *2000 American Institute of Physics.* [S0021-9606(00)52333-2]

I. INTRODUCTION

One of the most challenging problems in computer simulation of molecular systems is to evaluate the long-range electrostatic interaction between charged particles.[1] With periodic boundary conditions, the total electrostatic energy of a system of N particles is

$$E=\frac{1}{2}\sum_{\mathbf{n}}{}'\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{|\mathbf{r}_i-\mathbf{r}_j+L\mathbf{n}|} \tag{1}$$

(assuming the simulation box is cubic), where q_i and $\mathbf{r}_i$ are the charge and position of particle i, L is the size of the simulation box, $\mathbf{n}=(n_1,n_2,n_3)$, and the prime indicates that the $i=j$ terms are omitted when $\mathbf{n}=\mathbf{0}$.[2] The Ewald summation method has been widely used to handle the lattice sum in Eq. (1). In the method, the above-mentioned conditionally convergent series is written as a sum of a constant $E^{(0)}$ and two rapidly convergent series, the real space sum $E^{(r)}$ and the reciprocal space sum $E^{(k)}$,

$$E=E^{(0)}+E^{(r)}+E^{(k)}, \tag{2}$$

where

$$E^{(0)}=-\frac{\alpha}{\pi^{1/2}}\sum_{j=1}^{N}q_j^2, \tag{3}$$

$$E^{(r)}=\frac{1}{2}\sum_{\mathbf{n}}{}'\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j\ \mathrm{erfc}(\alpha|\mathbf{r}_i-\mathbf{r}_j+L\mathbf{n}|)}{|\mathbf{r}_i-\mathbf{r}_j+L\mathbf{n}|}, \tag{4}$$

$$E^{(k)}=\frac{1}{2\pi L}\sum_{\mathbf{n}\neq\mathbf{0}}\frac{1}{|\mathbf{n}|^2}\times\exp\left(-\frac{\pi^2|\mathbf{n}|^2}{L^2\alpha^2}\right)\left|\sum_{j=1}^{N}q_j\exp\left(\frac{2\pi i}{L}\mathbf{n}\cdot\mathbf{r}_j\right)\right|^2, \tag{5}$$

and α is a positive parameter chosen for computational efficiency.[2,3] For large systems, it has been shown that with optimal values of α, the complexity of the Ewald summation method is $O(N^{3/2})$.[4] For further details of the Ewald method we refer the reader to Refs. 2–5.

Much effort has recently been devoted to improving the efficiency of the Ewald summation method and developing alternative methods for large systems.[1,5] These include the fast multipole method[6–9] and the particle–particle and particle–mesh method.[10–12] In particular, the particle–mesh Ewald method[13,14] has been developed to speed up the computation of the reciprocal space part $E^{(k)}$ of the Ewald summation.

In what follows, we present a multipole method to speed up the computation of the real space part $E^{(r)}$ of the Ewald summation. The method is based on the techniques of the Ewald method and hierarchical tree codes.[6,15–20] It involves (i) direct evaluation of the reciprocal space sum; (ii) tree construction to subdivide the particles into clusters; (iii) multipole expansion to approximate the real space interaction between a particle and distant clusters; and (iv) a divide-and-conquer strategy to evaluate the real space interactions $E^{(r)}$ recursively.

II. MULTIPOLE APPROXIMATION

In this section, we develop the multipole approximation for $\phi(\mathbf{x})=(\sqrt{\pi}/2)\mathrm{erfc}(|\mathbf{x}|)/|\mathbf{x}|$. Consider the Taylor expansion of $\phi(\mathbf{x})$ about $\overline{\mathbf{x}}$,

$$\phi(\mathbf{x})=\sum_{\|\mathbf{k}\|=0}^{\infty}\frac{1}{\mathbf{k}!}D_{\mathbf{x}}^{\mathbf{k}}\phi(\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{k}}, \tag{6}$$

where $\mathbf{k}!=k_1!k_2!k_3!$, $\|\mathbf{k}\|=k_1+k_2+k_3$, $(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{k}}=(x_1-\overline{x_1})^{k_1}(x_2-\overline{x_2})^{k_2}(x_3-\overline{x_3})^{k_3}$, $D_{\mathbf{x}}^{\mathbf{k}}=\partial^{\|\mathbf{k}\|}/(\partial x_1{}^{k_1}\partial x_2{}^{k_2}\partial x_3{}^{k_3})$. Referring to Fig. 1, consider a particle i in a cluster A and a distant particle j. Let $\mathbf{x}_i=\alpha\mathbf{r}_i$ and $\mathbf{x}_j=\alpha\mathbf{r}_j$ be the scaled positions of particles i and j and $\mathbf{x}_A=\alpha\mathbf{r}_A$ be the scaled center of the cluster A. In Eq. (6), let $\mathbf{x}=\mathbf{x}_i-\mathbf{x}_j$ and $\overline{\mathbf{x}}=\mathbf{x}_A-\mathbf{x}_j$, we have the pth order Taylor approximation for the real space interaction between particle j and cluster A,

$$E_{j,A}^{(r)}=\sum_{i\in A}\frac{q_i q_j\ \mathrm{erfc}(\alpha|\mathbf{r}_i-\mathbf{r}_j|)}{|\mathbf{r}_i-\mathbf{r}_j|}=\frac{2\alpha}{\sqrt{\pi}}\sum_{i\in A}q_i q_j\phi(\mathbf{x}_i-\mathbf{x}_j) \tag{7}$$

$$\approx\frac{2\alpha}{\sqrt{\pi}}q_j\sum_{\|\mathbf{k}\|=0}^{p}\frac{1}{\mathbf{k}!}D_{\mathbf{x}}^{\mathbf{k}}\phi(\mathbf{x}_A-\mathbf{x}_j)\sum_{i\in A}q_i(\mathbf{x}_i-\mathbf{x}_A)^{\mathbf{k}}=\frac{2\alpha}{\sqrt{\pi}}q_j\sum_{\|\mathbf{k}\|=0}^{p}a_{\mathbf{k}}\,m_A^{\mathbf{k}}, \tag{8}$$

[a]Electronic mail: zduan@math.lsa.umich.edu

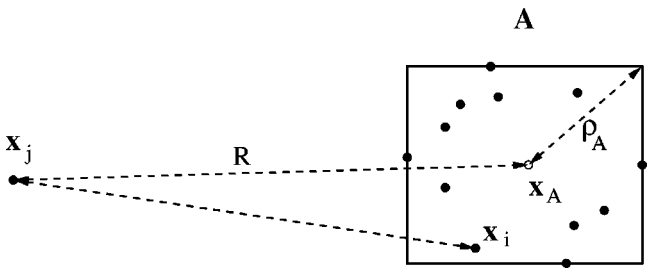

FIG. 1. Particle j and a cluster of particles A.

where $a_{\mathbf{k}}=(1/\mathbf{k}!)\,D_{\mathbf{x}}^{\mathbf{k}}\phi(\mathbf{x}_A-\mathbf{x}_j)$ is the Taylor coefficient and $m_A^{\mathbf{k}}=\Sigma_{\mathbf{x}_i\in A}q_i(\mathbf{x}_i-\mathbf{x}_A)^{\mathbf{k}}$ is the $\mathbf{k}$th multipole moment of cluster A. This equation computes the interaction between the particle j and the multipoles of cluster A.[21] The force exerted on particle j is the negative gradient of the potential at $\mathbf{r}_j$,

$$\mathbf{F}_{j,A}^{(r)}\approx-\frac{2\alpha^2}{\sqrt{\pi}}q_j\sum_{\|\mathbf{k}\|=0}^{p}(\nabla_{\mathbf{x}_j}a_{\mathbf{k}})m_A^{\mathbf{k}}$$
$$=\frac{2\alpha^2}{\sqrt{\pi}}q_j\sum_{\|\mathbf{k}\|=0}^{p}\begin{pmatrix}(k_1+1)a_{\mathbf{k}+\mathbf{e}_1}\\(k_2+1)a_{\mathbf{k}+\mathbf{e}_2}\\(k_3+1)a_{\mathbf{k}+\mathbf{e}_3}\end{pmatrix}m_A^{\mathbf{k}},\tag{9}$$

where $\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3$ are the standard Cartesian basis vectors. We note that the Taylor coefficients $a_{\mathbf{k}}$ in Eqs. (8) and (9) are independent of the number of particles in cluster A. Therefore, once the moments of each cluster $m_A^{\mathbf{k}}$ are obtained, the real space potential and the force at $\mathbf{r}_j$ can be computed cluster by cluster.

Next we derive the recurrence relations for fast computing the Taylor coefficients $a_{\mathbf{k}}$. Consider another function $\psi(\mathbf{x})=\frac{1}{2}e^{-|\mathbf{x}|^2}$. Obviously,

$$\frac{\partial\psi(\mathbf{x})}{\partial x_1}+2x_1\psi(\mathbf{x})=0.\tag{10}$$

Using the Leibniz rule for differentiating a product, we have

$$D_{\mathbf{x}}^{\mathbf{k}}\psi(\mathbf{x})+2x_1D_{\mathbf{x}}^{\mathbf{k}-\mathbf{e}_1}\psi(\mathbf{x})+2(k_1-1)D_{\mathbf{x}}^{\mathbf{k}-2\mathbf{e}_1}\psi(\mathbf{x})=0.\tag{11}$$

Two more equations can be obtained by permuting the indices. Now define $b_{\mathbf{k}}=(1/\mathbf{k}!)\,D_{\mathbf{x}}^{\mathbf{k}}\psi(\mathbf{x})$. Dividing Eq. (11) and the two others by $\mathbf{k}!$, we have the recurrence relations for the Taylor coefficients of ψ,

$$b_{\mathbf{k}}+\frac{2}{k_j}x_jb_{\mathbf{k}-\mathbf{e}_j}+\frac{2}{k_j}b_{\mathbf{k}-2\mathbf{e}_j}=0,\quad j=1,2,3,\tag{12}$$

where $b_{\mathbf{k}}=0$ when any of the indices is negative.

Now we shall follow the same procedure to obtain the recurrence relation for the Taylor coefficients of ϕ. It can be easily verified that

$$|\mathbf{x}|^2\frac{\partial\phi(\mathbf{x})}{\partial x_1}+x_1\phi(\mathbf{x})=\frac{\partial\psi(\mathbf{x})}{\partial x_1}.\tag{13}$$

Using the Leibniz rule again, we have

$$|\mathbf{x}|^2D_{\mathbf{x}}^{\mathbf{k}}\phi(\mathbf{x})+2\sum_{i=1}^{3}k_ix_iD_{\mathbf{x}}^{\mathbf{k}-\mathbf{e}_i}\phi(\mathbf{x})$$
$$+\sum_{i=1}^{3}k_i(k_i-1)D_{\mathbf{x}}^{\mathbf{k}-2\mathbf{e}_i}\phi(\mathbf{x})-x_1D_{\mathbf{x}}^{\mathbf{k}-\mathbf{e}_1}\phi(\mathbf{x})$$
$$-(k_1-1)D_{\mathbf{x}}^{\mathbf{k}-2\mathbf{e}_1}\phi(\mathbf{x})=D_{\mathbf{x}}^{\mathbf{k}}\psi(\mathbf{x})\tag{14}$$

as well as two more equations, which can be obtained by first differentiating $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ with respect to x_2 and x_3, respectively. Dividing Eq. (14) and the two others by $\mathbf{k}!$, recalling $a_{\mathbf{k}}=(1/\mathbf{k}!)\,D_{\mathbf{x}}^{\mathbf{k}}\phi(\mathbf{x})$, and simplifying, we have

$$|\mathbf{x}|^2a_{\mathbf{k}}+2\sum_{i=1}^{3}x_ia_{\mathbf{k}-\mathbf{e}_i}+\sum_{i=1}^{3}a_{\mathbf{k}-2\mathbf{e}_i}-\frac{x_j}{k_j}a_{\mathbf{k}-\mathbf{e}_j}-\frac{1}{k_j}a_{\mathbf{k}-2\mathbf{e}_j}$$
$$=b_{\mathbf{k}},\quad j=1,2,3.\tag{15}$$

Now multiplying Eq. (15) for $j=1,2,3$ by k_1, k_2, and k_3, respectively, summing the results, and collecting terms, we have the recurrence relation for $a_{\mathbf{k}}$,

$$\|\mathbf{k}\|\,|\mathbf{x}|^2a_{\mathbf{k}}+(2\|\mathbf{k}\|-1)\sum_{i=1}^{3}x_ia_{\mathbf{k}-\mathbf{e}_i}+(\|\mathbf{k}\|-1)\sum_{i=1}^{3}a_{\mathbf{k}-2\mathbf{e}_i}$$
$$=\|\mathbf{k}\|b_{\mathbf{k}},\tag{16}$$

where $a_{\mathbf{k}}=0$ when any of the indices is negative. For computational efficiency, we obtain the following recurrence relations from Eq. (15)

$$\frac{x_i}{k_i}a_{\mathbf{k}-\mathbf{e}_i}+\frac{1}{k_i}a_{\mathbf{k}-2\mathbf{e}_i}=\frac{x_j}{k_j}a_{\mathbf{k}-\mathbf{e}_j}+\frac{1}{k_j}a_{\mathbf{k}-2\mathbf{e}_j},\quad i,j=1,2,3.\tag{17}$$

In practice, $a_{\mathbf{k}}$ is computed in the same order as $\|\mathbf{k}\|$. Equation (17) is used together with Eq. (16) for boundary cases.

III. TREE CONSTRUCTION AND RECURSIVE FUNCTIONS

The tree used here is the same as described in Ref. 20. It consists of a hierarchy of nodes (also referred to as boxes or clusters). The root node is the smallest rectangular box containing all the particles in the center simulation box. The root is subdivided in each coordinate direction into a total of eight children. Before any further subdivision, each child is shrunk to the smallest rectangular box containing its particles. The shrunken boxes define the next level of nodes in the tree. The subdivision/shrinking continues until the number of particles in a node is less than or equal to a specified value N_0. These nodes form the leaves of the tree. Several bookkeeping steps are performed during the tree construction. The scaled particle positions and weights are stored in a global array in such a way that the members of a cluster appear in consecutive array locations. Several attributes associated with a node are also computed including the particle moments up to a chosen order p, as well as the scaled center $\mathbf{x}_A$ and radius ρ_A (see Fig. 1).

Having constructed the tree, the potential and force on a particle j are computed by traversing the tree.[16] There are three options at each step of the traverse: computing $E_{jA}^{(r)}$ and

```
compute_in_node(j, A)
1 if (j is in A)
2     if (A is a leaf)
3         compute E_jA^(r) and F_jA^(r) using the direct method;
4     else
5         for i = 1 to 8
6             compute_in_node(j, A.child[i]);
7 else
8     compute_out_node(j, A);

compute_out_node(j, A)
1 if (A is a leaf)
2     compute E_jA^(r) and F_jA^(r) using the direct method;
3 else
4     if (j and A are well separated)
5         compute E_jA^(r) and F_jA^(r) using a multipole approximation;
6     else
7         for i = 1 to 8
8             compute_out_node(j, A.child[i]);
```

FIG. 2. The two recursive functions, *compute_in_node* and *compute_out_node*.

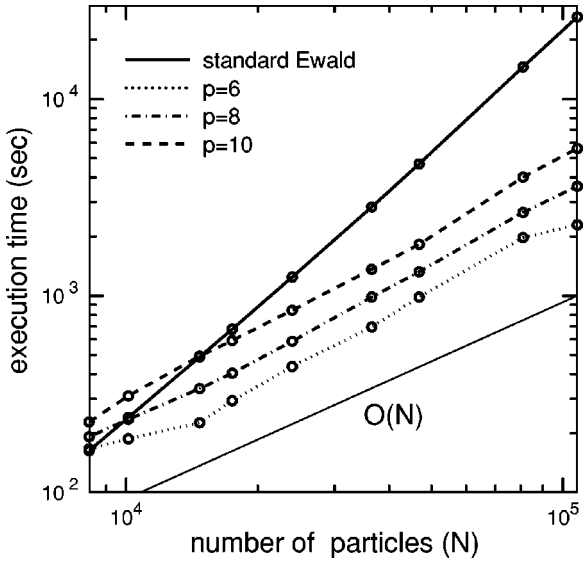

FIG. 3. Execution time required by the standard Ewald and the new method with the order of the approximation $p=6$, 8, 10 for the computation of Coulomb potential energy V and the forces $\mathbf{F}$.

$\mathbf{F}_{jA}^{(r)}$ using the direct method, using the multipole approximation, or descending the tree. The decision between these depends on the approximation acceptance criterion. The complete procedure is described in Fig. 2 and implemented using two recursive functions, *compute_in_node(j,A)* and *compute_out_node(j, A)*. The initial calls are *compute_in_node(j, root)* and *compute_out_node(j, replica_images)*. We say that j and A are well-separated if the ratio $\rho_A/R \leqslant s$, where R is the distance between the particle j and the center of cluster A and s is a separation parameter to be chosen for computational efficiency.

IV. NUMERICAL RESULTS

The Ewald summation based multipole method presented here was implemented in the C programming language and tested on a 296 MHz Sun UltraSPARC-II workstation. The standard Ewald summation method in double precision is used as the benchmark for testing the performance of the new method. In the implementation of the Ewald method, the potential is computed using the expressions given in Eq. (2) and the forces are evaluated using the expressions given in Ref. 5. The summation in real space is carried out using the linked-cell method with a cutoff of radius r_c.[10] The reciprocal space summation is done using a cutoff of radius n_c. The complementary error function $\mathrm{erfc}(x)$ is evaluated using power series for small arguments and the asymptotic expansion for large arguments.[22] The potential and forces obtained using the standard Ewald summation with $\alpha=6/L$ and very large cutoff radius $r_c=L$ and $n_c=12$ are used as correct results for testing the accuracy of the method. To demonstrate the performance, a set of commonly used parameters $\alpha=5.6/L$, $r_c=L/2$, $n_c=6$ is used.[2,23] We note that with $\alpha=5.6/L$, (i) the number of operations in the reciprocal space computation is of order $O(N)$; (ii) only the center simulation box and its nearest neighbors are needed for the real space computation, but the complexity is $O(N^2)$.

In the implementation of the new method, the maximum number of particles in the leaf clusters (N_0) is chosen to be 20 and the separation parameter s is taken to be 0.5. Currently, they are selected on a trial and error basis. $\alpha=5.6/L$ and the same cutoffs $r_c=L/2$ and $n_c=6$ are used. The computation in the reciprocal space is done by the same code used for the Ewald method.

The test data are a set of water molecules. The TIP4P water model[24] was used and a 1.6 ps molecular dynamics simulation was performed to generate the configurations of the water molecules.[25] The electrostatic potential and forces are computed based on this set of data. The numerical results for three values of p are shown in Figs. 3 and 4. The relative error in the force is computed using

$$\left(\frac{\sum_{i=1}^{N} |\mathbf{F}_i - \hat{\mathbf{F}}_i|^2}{\sum_{i=1}^{N} |\mathbf{F}_i|^2} \right)^{1/2}, \tag{18}$$

where $\mathbf{F}_i$ is the correct force and $\hat{\mathbf{F}}_i$ is the approximate result obtained by either the standard Ewald or the new method.

The following observations can be made from the numerical results. (i) For large values of N, the new method is significantly faster than the standard Ewald method. The speed-up is about 7.3 for $p=8$ and $N=107{,}811$. (ii) The break-even point between the two methods is below $N=10\,000$ for $p=6$ and $p=8$. It is about 15 000 for $p=10$. (iii) The observed CPU time complexity of the new method is consistent with $O(N \log N)$, as expected for a particle-cluster tree code.[16] (iv) With $p=8$, the error for the new method is slightly higher than the error for standard Ewald, but with $p=10$, the error is lower. (v) The relative errors in the forces for the new method is $O(s^p)$ as p increases; this can be seen in Fig. 4 with $s=0.5$ and was also obtained for $s=0.4$ and $s=0.6$ (data not shown). (vi) The relative error in

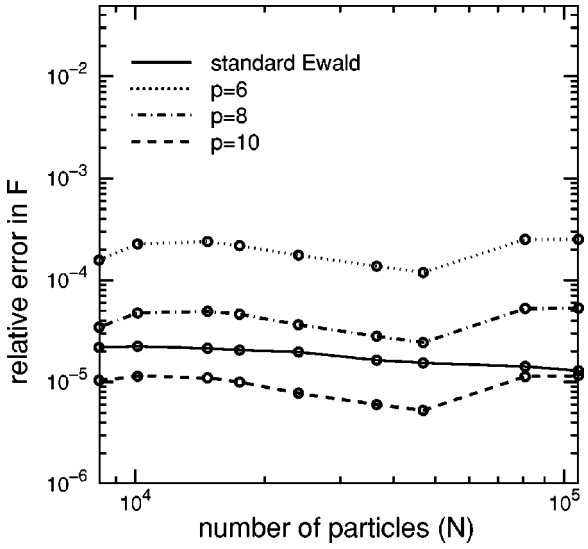

FIG. 4. Relative error in Coulomb force **F** computed by the standard Ewald and the new method with the order of the approximation $p=6$, 8, 10.

the Coulomb potential energy, $|E-\hat{E}|/E$, is about 2 orders smaller than that in the forces for both methods (data not shown).

V. FINAL REMARK

The aim of this work was to present a new method to speed up the computation of the real space part of the Ewald summation. No attempt was made to optimize the parameter α. As mentioned previously, with the value of α used here the computation cost is $O(N)$ in the reciprocal space and $O(N^2)$ in the real space for the standard Ewald method. The new method reduces the $O(N^2)$ complexity to be $O(N \log N)$. The idea of the popular particle–mesh Ewald method[13] is to choose α large enough so that a fixed cutoff radius can be applied to reduce the computation cost of the real space sum to be $O(N)$. The reciprocal space sum is then computed using fast Fourier transformation with a complexity $O(N \log N)$. Since the Ewald summation technique yields two distinct computational problems—the evaluation of the real space sum and the reciprocal space sum, the new method can be combined with the particle–mesh Ewald method to reduce the constant of proportionality. The combined method can be expected to perform very well on a variety of computer systems by tuning the parameter α.

ACKNOWLEDGMENTS

This work was supported by a University of Michigan Rackham Faculty Fellowship and NSF Grant Nos. DMS-9506452 and DMS-9973293. The authors thank Wei-Li Qian for carefully reading the manuscript and providing valuable comments, and Samuel Krimm and his research group for helpful discussions.

[1] T. Schlick, R. D. Skeel, A. T. Brunger, L. V. Kale, J. A. Board, Jr., J. Hermans, and K. Schulten, J. Comput. Phys. **151**, 9 (1999).
[2] M. P. Allen and D. J. Tildesley, *Computer Simulaton of Liquids* (Oxford University Press, New York, 1987).
[3] S. W. de Leeuw, J. W. Perram, and E. R. Smith, Proc. R. Soc. London, Ser. A **373**, 27 (1980).
[4] S. W. de Leeuw, J. W. Perram, and H. G. Petersen, Mol. Phys. **65**, 875 (1988).
[5] A. Y. Toukmaji and J. A. Board, Jr., Comput. Phys. Commun. **95**, 73 (1996).
[6] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
[7] K. E. Schmidt and M. A. Lee, J. Stat. Phys. **63**, 1223 (1991).
[8] H.-Q. Ding, N. Karasawa, and W. A. Goddard III, Chem. Phys. Lett. **196**, 6 (1992).
[9] M. Challacombe, C. White, and M. Head-Gordon, J. Chem. Phys. **107**, 10131 (1997).
[10] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
[11] J. Shimada, H. Kaneko, and T. Takada, J. Comput. Chem. **14**, 867 (1993).
[12] B. A. Luty, M. E. Davis, I. G. Tironi, and W. F. van Gunsteren, Mol. Simul. **14**, 11 (1994).
[13] T. Darden, D. York, and L. Pedersen, J. Chem. Phys. **98**, 10089 (1993).
[14] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, J. Chem. Phys. **103**, 8577 (1995).
[15] A. W. Appel, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **6**, 85 (1985).
[16] J. Barnes and P. Hut, Nature (London) **324**, 446 (1986).
[17] L. Greengard, Science **265**, 909 (1994).
[18] C. Draghicescu and M. Draghicescu, J. Comput. Phys. **116**, 69 (1995).
[19] K. Lindsay, Ph.D. thesis, University of Michigan, Ann Arbor, 1997.
[20] Z.-H. Duan and R. Krasny (unpublished).
[21] A. J. Stone, *The Theory of Intermolecular Forces* (Clarendon, Oxford, 1996).
[22] S. Zhang and J. Jin, *Computation of Special Functions* (Wiley, New York, 1996).
[23] G. Hummer, L. R. Pratt, and A. E. Garcia, J. Phys. Chem. **102**, 7885 (1998).
[24] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, J. Chem. Phys. **79**, 926 (1983).
[25] D. C. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 1995).