A Treecode Algorithm for Computing Ewald Summation of Dipolar Systems

Zhong-Hui Duan
Department of Computer Science
University of Akron
Akron, OH 44325
zduan@cs.uakron.edu

Robert Krasny
Department of Mathematics
University of Michigan
Ann Arbor, MI 48109
krasny@umich.edu

ABSTRACT

We present a treecode algorithm for efficiently computing the real space part of Ewald summation in periodic dipolar systems. The algorithm uses multipole expansion in Cartesian coordinates to approximate the real space interaction between a dipole and a distant cluster of dipoles. The necessary Taylor coefficients are computed efficiently using recurrence relations. Two divide-and-conquer evaluation procedures are described. Test results are presented for systems of randomly generated dipoles.

Keywords

Ewald summation, fast electrostatics, tree method

1. INTRODUCTION

Along with the increasing use of computer simulations for biomolecular systems, developing and using more accurate and reliable governing force fields are becoming essential for advanced understanding of the complex systems. Particularly, much effort has been made on the development of force fields including many body effects such as polarizability. On the other hand, using polarizable force fields considerably increases the computational cost, which is prohibitively expensive for large scale and long time simulations [13]. In this paper, we present a treecode algorithm for efficiently computing electrostatic interactions in periodic dipolar systems.

In molecular dynamics simulations, periodic boundary conditions are commonly used to reduce the surface effects. Consider N point dipoles in a cubic simulation box that is replicated periodically in all directions. The electrostatic energy of such a system can be expressed as a sum of dipole interactions,

$$E = \frac{1}{2}\sum_{\mathbf{n}}{}' \sum_{i=1}^{N}\sum_{j=1}^{N}(\boldsymbol{\mu}_i \cdot \nabla_{\mathbf{r}_i})(\boldsymbol{\mu}_j \cdot \nabla_{\mathbf{r}_j})\frac{1}{|\mathbf{r}_i - \mathbf{r}_j + L\mathbf{n}|}, \quad (1)$$

SAC 2003, Melbourne, Florida, USA

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ are point dipoles at position $\mathbf{r}_i$ and $\mathbf{r}_j$, $\nabla_{\mathbf{r}_i}$ and $\nabla_{\mathbf{r}_j}$ denote differentiation at $\mathbf{r}_i$ and $\mathbf{r}_j$, L is the side length of the simulation box, the index $\mathbf{n} = (n_1, n_2, n_3)$ runs through the periodic images of the box, and the prime indicates that the $i = j$ terms are omitted when $\mathbf{n} = \mathbf{0}$ [1]. Although the dipolar potential converges faster than Coulomb potential, its long range slowly decaying nature still renders accurately evaluating the potential energy and force a challenging and important problem. To reduce the computational cost, as for Coulomb systems, the Ewald summation technique has often been used. It splits the direct summation of the interactions (Eq. (1)) into a sum of a constant $E^{(0)}$ and two rapidly convergent series, the real space sum $E^{(r)}$ and the reciprocal space sum $E^{(k)}$,

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

where

$$E^{(0)} = -\frac{2\alpha^3}{3\pi^{1/2}}\sum_{j=1}^{N}|\boldsymbol{\mu}_j|^2, \quad (3)$$

$$E^{(r)} = \frac{1}{2}\sum_{\mathbf{n}}{}'\sum_{i=1}^{N}\sum_{j=1}^{N}(\boldsymbol{\mu}_i \cdot \nabla_{\mathbf{r}_i})(\boldsymbol{\mu}_j \cdot \nabla_{\mathbf{r}_j})\cdot \frac{\operatorname{erfc}(\alpha|\mathbf{r}_i - \mathbf{r}_j + L\mathbf{n}|)}{|\mathbf{r}_i - \mathbf{r}_j + L\mathbf{n}|}, \quad (4)$$

$$E^{(k)} = \frac{1}{2\pi L}\sum_{\mathbf{n}\neq 0}\frac{1}{|\mathbf{n}|^2}\exp\Big(-\frac{\pi^2|\mathbf{n}|^2}{L^2\alpha^2}\Big)|\cdot \sum_{j=1}^{N}(\boldsymbol{\mu}_j \cdot \nabla_{\mathbf{r}_j})\exp(\frac{2\pi i}{L}\mathbf{n}\cdot\mathbf{r}_j)|^2, \quad (5)$$

and α is a positive parameter chosen for computational efficiency [1, 5]. The complementary error function in the real space sum and the exponential function in the reciprocal space sum decay rapidly with the index $\mathbf{n}$, therefore cutoffs r_c, k_c can be used to compute the Ewald sum, i.e. only terms satisfying $|\mathbf{r}_i - \mathbf{r}_j + L\mathbf{n}| \leq r_c$ for $E^{(r)}$ and $|\mathbf{n}| \leq k_c$ for $E^{(k)}$ are retained for the computation. The magnitude of the Ewald parameter α controls the relative rates of convergence of the real space sum and the reciprocal space sum. When α is large, $E^{(r)}$ converges rapidly and can be evaluated to a given accuracy in $O(N)$ operations using an appropriate cutoff r_c; however in this case $E^{(k)}$ converges slowly and $O(N^2)$ operations are required since the cutoff k_c must be large enough to attain the desired accuracy. The situation is reversed when α is small, and therefore in either case, the classical Ewald method requires $O(N^2)$ operations. The cost can be reduced to $O(N^{3/2})$ by optimizing the parameters α, r_c, k_c as a function of N [19, 12]. The hidden constant in front of $N^{3/2}$ can be further reduced using the linked-cell method to reduce the computational cost of locating the dipoles which are within the cutoff radius of a given dipole [19, 10]. A particle-mesh Ewald based method (PME) of order $O(N \log N)$ is also reported for dipole interactions [17]. The complexity reduction in PME is accomplished by choosing a large value for α; the real space sum is computed in $O(N)$ and the cost of evaluating the reciprocal space sum is reduced from $O(N^2)$ to $O(N \log N)$ using a particle-mesh interpolation procedure and the fast Fourier transform. For further details of the Ewald method we refer the reader to Refs. [1, 5, 19, 12, 17, 4, 18].

The present work concerns an alternative class of methods, treecode algorithms, that rely on multipole expansions. In a treecode algorithm, particles are divided into nested clusters and the interaction between a particle and a distant cluster of particles is approximated using a multipole expansion. The first treecode algorithms used a monopole approximation for the particle-cluster interaction and a divide-and-conquer strategy to choose the clusters [2, 3]. Greengard and Rohklin's fast multipole method (FMM) is a more elaborate procedure that uses a higher order multipole approximation and a technique for evaluating the approximation by converting it to a local series [8, 9, 14]. These methods reduce the operation count to $O(N \log N)$ or $O(N)$. Extension of FMM to dipolar systems has also been made [11].

In this paper, we present a simple treecode algorithm for computing the real-space part of the Ewald summation for dipolar systems. Our approach involves (i) direct evaluation of the reciprocal space sum; (ii) tree construction to subdivide the dipoles into clusters; (iii) multipole expansion to approximate the real space interaction between a dipole and a distant cluster of dipoles; (iv) divide-and-conquer strategy to evaluate the real space interactions recursively. In what follows, first we describe the multipole approximation for dipole-cluster interactions. We then present the construction of the tree and two divide-and-conquer strategies to apply the multipole approximation recursively. In section **5**, we give the numerical results obtained for systems of randomly generated dipoles. Finally, we summarize our findings in section **6**.

2. MULTIPOLE APPROXIMATION

In this section, we describe the multipole approximation for the dipole interaction between a dipole and a cluster of dipoles. Define the potential function $\phi(\mathbf{x}) = \frac{\sqrt{\pi}}{2}\mathrm{erfc}(|\mathbf{x}|)/|\mathbf{x}|$. 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 $\mathbf{k}! = k_1!k_2!k_3!$, $\|\mathbf{k}\| = k_1 + k_2 + k_3$, $(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{k}} = (x_1 - \overline{x}_1)^{k_1}(x_2 - \overline{x}_2)^{k_2}(x_3 - \overline{x}_3)^{k_3}$, $D_{\mathbf{x}}^{\mathbf{k}} = \partial^{\|\mathbf{k}\|}/\partial x_1{}^{k_1}\partial x_2{}^{k_2}\partial x_3{}^{k_3}$. Consider a point dipole $\boldsymbol{\mu}_i$ at $\mathbf{r}_i$ in a cluster A and a distant dipole $\boldsymbol{\mu}_j$ at $\mathbf{r}_j$. Let $\mathbf{p}_i = \alpha\boldsymbol{\mu}_i$ and $\mathbf{p}_j = \alpha\boldsymbol{\mu}_j$ be the scaled dipoles and $\mathbf{x}_i = \alpha\mathbf{r}_i$ and $\mathbf{x}_j = \alpha\mathbf{r}_j$ be the scaled positions, and $\mathbf{x}_A = \alpha\mathbf{r}_A$ be the scaled center of the cluster A as in Figure 1.

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

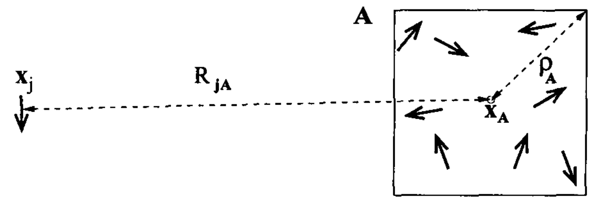

Figure 1: Dipole j and a cluster of dipoles A. $\mathbf{x}_A$ is the scaled cluster center; ρ_A is the scaled cluster radius; R_{jA} is the distance between the dipole and the cluster center.

pth order Taylor approximation for the real space interaction between dipole j and cluster A,

$$\begin{aligned} E_{j,A}^{(r)} &= \sum_{i\in A}(\boldsymbol{\mu}_i \cdot \nabla_{\mathbf{r}_i})(\boldsymbol{\mu}_j \cdot \nabla_{\mathbf{r}_j})\frac{\mathrm{erfc}(\alpha|\mathbf{r}_i - \mathbf{r}_j|)}{|\mathbf{r}_i - \mathbf{r}_j|} &(7)\\ &= \frac{2\alpha}{\sqrt{\pi}}\sum_{i\in A}(\mathbf{p}_i \cdot \nabla_{\mathbf{x}_i})(\mathbf{p}_j \cdot \nabla_{\mathbf{x}_j})\phi(\mathbf{x}_i - \mathbf{x}_j) &(8)\\ &\approx \frac{2\alpha}{\sqrt{\pi}}\sum_{\|\mathbf{k}\|=0}^{p}(\mathbf{p}_j \cdot \nabla_{\mathbf{x}_j})\frac{1}{\mathbf{k}!}D_{\mathbf{x}}^{\mathbf{k}}\phi(\mathbf{x}_A - \mathbf{x}_j)\cdot \\ &\qquad \sum_{i\in A}(\mathbf{p}_i \cdot \nabla_{\mathbf{x}_i})(\mathbf{x}_i - \mathbf{x}_A)^{\mathbf{k}} &(9)\\ &= \frac{2\alpha}{\sqrt{\pi}}\sum_{\|\mathbf{k}\|=0}^{p}(\mathbf{p}_j \cdot \nabla_{\mathbf{x}_j})a_{\mathbf{k}}\, m_A^{\mathbf{k}}, &(10) \end{aligned}$$

where $a_{\mathbf{k}} = \frac{1}{\mathbf{k}!}D_{\mathbf{x}}^{\mathbf{k}}\phi(\mathbf{x}_A - \mathbf{x}_j)$ is the Taylor coefficient and $m_A^{\mathbf{k}} = \sum_{i\in A}(\mathbf{p}_i \cdot \nabla_{\mathbf{x}_i})(\mathbf{x}_i - \mathbf{x}_A)^{\mathbf{k}}$ is the **k**th multipole moment of cluster A. The last equation computes the electrostatic interaction between dipole j and the multipoles of cluster A. [16]. The force exerted on dipole j is the negative gradient of the potential energy at $\mathbf{r}_j$,

$$\begin{aligned} \mathbf{F}_{j,A}^{(r)} &\approx -\frac{2\alpha^2}{\sqrt{\pi}}\sum_{\|\mathbf{k}\|=0}^{p}\nabla_{\mathbf{x}_j}(\mathbf{p}_j \cdot \nabla_{\mathbf{x}_j})a_{\mathbf{k}}\; m_A^{\mathbf{k}} &(11)\\ &= \frac{2\alpha^2}{\sqrt{\pi}}\sum_{\|\mathbf{k}\|=0}^{p}(\mathbf{p}_j \cdot \nabla_{\mathbf{x}_j})\begin{pmatrix}(k_1+1)a_{\mathbf{k}+\mathbf{e}_1}\\(k_2+1)a_{\mathbf{k}+\mathbf{e}_2}\\(k_3+1)a_{\mathbf{k}+\mathbf{e}_3}\end{pmatrix} m_A^{\mathbf{k}}, &(12) \end{aligned}$$

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. (10) and (12) are independent of the number of dipoles in cluster A. Therefore, once the moments of each cluster $m_A^{\mathbf{k}}$ are obtained, the real space potential energy and the force at $\mathbf{r}_j$, $E_j^{(r)}$ and $\mathbf{F}_j^{(r)}$, can be computed cluster by cluster. In practice, it is more efficient to employ the approximation only when the multipole acceptance criterion is satisfied (detailed in section **4**).

Recurrence Relations. Explicit formulas for the Taylor coefficients $\mathbf{a}_{\mathbf{k}}$ can be developed, but we found it simpler and more efficient to use recurrence relations instead. Consider another function $\psi(\mathbf{x}) = \frac{1}{2}e^{-|\mathbf{x}|^2}$ and define $b_{\mathbf{k}} = \frac{1}{\mathbf{k}!}D_{\mathbf{x}}^{\mathbf{k}}\psi(\mathbf{x})$. It is easy to verify that $b_{\mathbf{k}}$ satisfies the following very simple recurrence relations

$$b_{\mathbf{k}} + \frac{2}{k_j}x_j b_{\mathbf{k}-\mathbf{e}_j} + \frac{2}{k_j}b_{\mathbf{k}-2\mathbf{e}_j} = 0, \qquad j = 1, 2, 3, \tag{13}$$

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

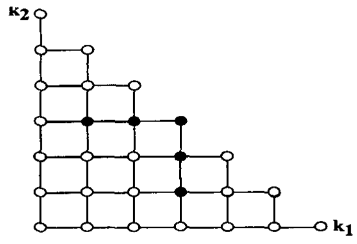

Figure 2: Stencil of the Recurrence Relation (14) in two Dimensions.

derivation of the recurrence relations for the Taylor coefficients $\mathbf{a_k}$ is also straightforward using Leibniz's rule for differentiating the product of two functions:

$$\|\mathbf{k}\||\mathbf{x}|^2 a_{\mathbf{k}} + (2\|\mathbf{k}\| - 1)\sum_{i=1}^{3} x_i a_{\mathbf{k}-\mathbf{e}_i} + (\|\mathbf{k}\| - 1)\sum_{i=1}^{3} a_{\mathbf{k}-2\mathbf{e}_i} = \|\mathbf{k}\| b_{\mathbf{k}}, \qquad (14)$$

where $a_{\mathbf{k}} = 0$ when any of the indices is negative. Figure 2 shows the associated stencil in two dimensions. To reduce the terms involved in the recurrence relations, equation (13), (14) and a slightly different alternative form of equation (14) are used in practice[6].

3. TREE CONSTRUCTION

The tree used here is a variant of the Barnes-Hut tree. The tree construction procedure divides the particles into a collection of nested clusters (also referred to as cells or nodes). The root node is the cubic box containing all the dipoles in the center simulation box. The root is subdivided in each coordinate direction into a total of eight children. The children define the next level of nodes in the tree. The subdivision continues until the number of dipoles in a node is less than or equal to a specified value N_0. These nodes form the leaves of the tree. Bookkeeping steps are performed during the tree construction. The scaled dipole positions, directions and strength 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 multipole moments up to a chosen order p, as well as the scaled center $\mathbf{x}_A$ and radius ρ_A as in Figure 1.

4. EVALUATION PROCEDURES

Having constructed the tree, the potential energy and force on a dipole are computed by traversing the tree. We present here two different divide-and-conquer strategies to accomplish the task. The first one is shown in Figure 3. It cycles through the dipoles and computes the interaction between a dipole j and a cluster of dipoles A recursively. Interaction between a dipole and a leaf cluster of dipoles is computed directly using the Ewald method. The multipole approximation is performed when the dipole is *well-separated* from a nonleaf cluster, i.e. the following multipole acceptance criterion (MAC) is satisfied,

$$\rho_A / R_{jA} \le s \qquad (15)$$

```
dipole_cluster_procedure()
for $j = 1$ to $N$
  compute_in_node($j$, root)
  /*compute energy & force due to the 26 nearest images of root*/
  for $i = 1$ to 26
    compute_out_node($j$, $image_i$)

compute_in_node($j$, $A$)
if ($j$ is in $A$)
  if ($A$ is a leaf)
    compute $E_{jA}^{(r)}$ and $\mathbf{F}_{jA}^{(r)}$ using direct Ewald;
  else /*examine the 8 children of cluster A*/
    for $i = 1$ to 8
      compute_in_node($j$, $A.child[i]$);
else
  compute_out_node($j$, $A$);

compute_out_node($j$, $A$)
if ($A$ is a leaf)
  compute $E_{jA}^{(r)}$ and $\mathbf{F}_{jA}^{(r)}$ using direct Ewald;
else
  if ($j$ and $A$ are well separated)
    compute $E_{jA}^{(r)}$ and $\mathbf{F}_{jA}^{(r)}$ using a multipole approximation;
  else
    for $i = 1$ to 8
      compute_out_node($j$, $A.child[i]$);
```

Figure 3: Outline of the dipole-cluster evaluation procedure and the two recursive functions involved.

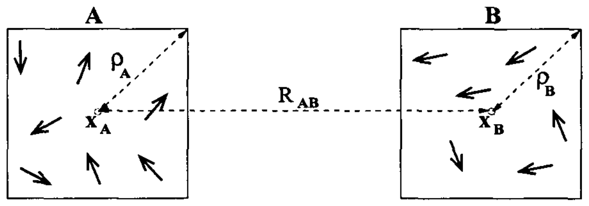

Figure 4: Two clusters A, B of dipoles. $\mathbf{x}_A$, $\mathbf{x}_B$ are the cluster centers; ρ_A, ρ_B are the cluster radii; R_{AB} is the distance between the cluster centers.

where ρ_A, R_{jA} are the cluster radius and dipole-cluster distance defined in Figure 1. s is a user-specified parameter for controlling the computational accuracy. If the MAC is not satisfied, the eight children of the given cluster are examined; this procedure is continued until a leaf cluster is encountered or the MAC is satisfied.

The second strategy is shown in Figure 5. It cycles through all the clusters of dipoles. If two clusters are leaves, direct Ewald summation is performed using the symmetry property of pairwise interactions. The multipole acceptance criterion is defined for two clusters A and B which are not both leaves [7],

$$(\rho_A + \rho_B)/R_{AB} \le s \qquad (16)$$

where ρ_A, ρ_B are the cluster radii and R_{AB} is the distance between the centers of the two clusters defined in Figure 4. If the MAC is satisfied, the algorithm cycles through the dipoles in the two clusters and computes the interaction between a dipole in one cluster and another cluster of dipoles using multipole approximation. If the MAC is not satisfied, the eight children of the cluster of larger radius are examined; the procedure is continued until both clusters are leaves or the MAC is satisfied.

We refer to the first procedure as the dipole-cluster evaluation procedure and the second as the cluster-cluster evaluation procedure. The main difference between them is on

```
cluster_cluster_procedure()
compute_one_node( root)
/*compute energy & force due to the 26 nearest images of root*/
for $i = 1$ to 26
  compute_two_nodes(root, $image_i$)

compute_one_node($A$)
if ($A$ is a leaf)
  $\forall\, i, j \in A$
  compute $E_{ij}^{(r)}$ & $\mathbf{F}_{ij}^{(r)}$ by direct Ewald using pairwise symmetry;
else/*examine the 8 children of cluster A*/
  for $i = 1$ to 8
    compute_one_node(A.child[i]);
    for $j = i + 1$ to 8
      compute_two_nodes(A.child[i], A.child[j]);

compute_two_nodes($A, B$)
if (both $A$ and $B$ are leaves)
  $\forall\, i \in A,\ j \in B$
  compute $E_{ij}^{(r)}$ & $\mathbf{F}_{ij}^{(r)}$ by direct Ewald using pairwise symmetry;
else if ($A$ and $B$ are well separated)
  $\forall\, i \in A,\ j \in B$
  compute $E_{iB}^{(r)}$ & $\mathbf{F}_{iB}^{(r)}$; $E_{jA}^{(r)}$ & $\mathbf{F}_{jA}^{(r)}$ using multipole approximation;
else if (($A$ is a leaf) or (B is not a leaf and $r_B > r_A$))
  for $j = 1$ to 8
      compute_two_nodes(A, B.child[j]);
else
  for $i = 1$ to 8
      compute_two_nodes(A.child[i], B);
```

Figure 5: Outline of the cluster-cluster evaluation procedure and the two recursive functions involved.

how the MAC is checked. It is clear that the first MAC (eq. (15)) is easier to be satisfied and therefore more approximations are made if the dipole-cluster procedure is used. The second MAC (eq. (16)) is checked for two clusters of dipoles. When the direct Ewald method is used for leaf clusters, the symmetry property of pairwise interactions can be considered. Therefore only half of the computation for leaf clusters need to be carried out. The impact of these differences on the performance of the treecode is documented in Figure 8 and Figure 9 in section 5.

5. IMPLEMENTATION AND NUMERICAL RESULTS

The Ewald summation based multipole method for dipolar systems and the classical Ewald summation method with a neighbor cell list were implemented in the C programming language and tested on an 1133 MHz IBM ThinkPad. The classical Ewald summation method is used as the benchmark for testing the performance of the treecode. In the implementation of the classical Ewald method, the real space part was evaluated with a cutoff r_c and explicit formulas for the 1st and 2nd derivatives of $\operatorname{erfc}(\alpha|\mathbf{r}|)/|\mathbf{r}|$ obtained using Smith's recurrence formula [17, 15]. The complementary error function $\operatorname{erfc}(x)$ was evaluated using power series for small arguments and the asymptotic expansion for large arguments [21]. To reduce the cost of *neighbor finding*, i.e. locating the dipoles which are within the cutoff radius of a given dipole, a 3-D Hockney-Eastwood lattice of $(M_1 \times M_2 \times M_3)$ cells was used as a mesh to cover the center simulation box and the images of its nearest neighbors. Each of the cells in the central simulation box was linked with a list of cells within its cutoff radius. The contributions to the energy and force on a dipole in a cell A are only from dipoles within cell A and in the cells that are linked to A. Figure 6 depicts a mesh and a list of cells linked to cell

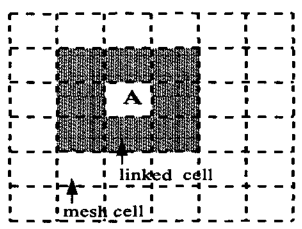

Figure 6: The simulation box and its neighbor images are divided into cells. Contributions to the energy and force on a dipole in cell A are only from the dipoles in A and the cells (shaded) linked to A.

```
main()
input dipole positions, directions and strength
input user-specified parameters
  $\alpha$ : Ewald parameter
  $r_c$ : real space cut-off radius
  $s$ : MAC parameter
  $p$ : order of approximation
  $N_0$ : maximum number of particles in a leaf
construct tree
compute real space sum using
    dipole_cluster_procedure() or cluster_cluster_procedure();
```

Figure 7: Implementation of the treecode algorithm.

A in two dimensions for the set of parameter values given in equation (18). The reciprocal space sum was computed with a cutoff n_c using the expression given in equation (5) with explicit differentiation of the exponential function.

One set of results was computed using the classical Ewald summation with parameter values

$$\alpha = 6/L,\ r_c = L,\ k_c = 12; \qquad (17)$$

these results are correct to double precision accuracy and they serve as a benchmark for determining the error. We compared the performance of the treecode algorithm and the classical Ewald method for parameter values

$$\alpha = 5/L,\ r_c = L/2,\ k_c = 4; \qquad (18)$$

these are commonly used values [1, 5] that provide moderate accuracy at much lower cost than the values in (17). We note that with this set of parameters, (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 of the classical Ewald computation is $O(N^2)$.

The implementation of the treecode algorithm is outlined in Figure 7. The real space cutoff was implemented by requiring that the particle-cluster interactions satisfy the criterion $|\mathbf{x}_j - \mathbf{x}_A| \le r_c + r_A$; i.e. a cluster A contributes to the real space sum as long as it overlaps the sphere of radius r_c centered at the particle $\mathbf{x}_j$. As a result, the treecode computation included more dipole interactions than those entering the classical Ewald computation. The neighbor finding procedure provided by the hierarchical tree data structure is slightly more efficient beyond the linked-cell method. It eliminates a cluster of dipoles with a single distance calculation when $|\mathbf{x}_j - \mathbf{x}_A| > r_c + r_A$ [20]. In addition, using the cluster-cluster evaluation procedure, two clusters of dipoles can be eliminated when $|\mathbf{x}_A - \mathbf{x}_B| > r_c + r_A + r_B$. It is interesting to recall that the volume of a sphere of radius r is $\frac{4}{3}\pi r^3 = 0.5236$ for $r = 1/2$. This suggests that if the dipoles are uniformly distributed in the simulation box, only about 53% of the dipoles are located within the inscribed sphere and contribute to the energy and force on the dipole located at the center of the box. We believe this is why when the cluster-cluster evaluation procedure is used, the treecode algorithm, even with the overhead of constructing the tree and computing the attributes for each node, outperformed the classical Ewald method for all values of N as shown in Figure 9. Because the pairwise symmetry property can not be utilized in the treecode with the particle-cluster evaluation procedure, the advantage of the neighbor finding approach is not apparent in the results shown in Figure 8.

The parameter values for the treecode algorithm were chosen to be representative rather than optimal. The maximum number (N_0) of dipoles in a leaf cluster was set to be 20 and the separation parameter s was taken to be 0.4 for the treecode using the dipole-cluster evaluation procedure. Since the leaf interaction can be computed more efficiently using the pairwise symmetry property when the cluster-cluster evaluation procedure is used, different values of $N_0 = 40$ and $s = 0.5$ were chosen for the treecode using the cluster-cluster evaluation procedure. The computation in the reciprocal space is done by the same code used for the classical Ewald method.

The test data are sets of randomly generated dipoles with a fixed density. The electrostatic potential energy and forces were computed based on the data. The numerical results for three values of p are shown in Figure 8 and Figure 9. The root mean square (rms) error in the force was computed using $\sqrt{\frac{1}{N}\sum_{i=1}^{N}|\mathbf{F}_i - \hat{\mathbf{F}}_i|^2}$, where $\mathbf{F}_i$ is the correct force and $\hat{\mathbf{F}}_i$ is the approximate result obtained by either the classical Ewald or the treecode method. We note that the rms error is more meaningful than relative error for randomly generated data; for example, randomly generated dipole positions could produce forces of very large magnitude at some dipoles.

The following observations can be made from the numerical results. (i) For large values of N, the treecode algorithm is significantly faster than the classical Ewald method; for example, with $N = 200,000$, the speed-up is about 4.9 and 4 for the two different evaluation procedures. The cross-over point between the classical Ewald method and the treecode with the dipole-cluster evaluation procedure is about $N = 12,000$ for $p = 4$, and 25,000 for $p = 8$. There is no cross-over point when the cluster-cluster evaluation procedure is used; the treecode is always faster than the Ewald method although most of the computation for $N \leq 10,000$ was performed using direct Ewald summation. As noted above, this is because the neighbor finding procedure for the treecode is slightly more efficient than that for the classical Ewald method. Apparently, the overhead of constructing the tree and computing the attributes associated with the tree nodes is negligible as compared with the cost of computing the energy and forces. (ii) As compared with the dipole-cluster evaluation procedure, the cluster-cluster evaluation procedure performs better for small to moderately large systems. However, as the system size increases, the difference diminishes and when $N = 200,000$, the algorithm with the dipole-cluster evaluation procedure surpasses the one with the cluster-cluster evaluation procedure. (iii) The observed CPU time complexity of the new method is consistent with $O(N \log N)$, as expected for a particle-cluster treecode [3]. (iv) The rms error in the force for the treecode algorithm is $O(s^p)$ as p increases. We note that the treecode error should be judged by how well it compares with the classical Ewald error, not by how small it is. Since the tree data structure is discrete, the number of dipoles in clusters at each level varies with N; for example, with $N_0 = 20$, the average number of dipoles in a leaf is $3.05 = 100,000/8^5$ for $N = 100,000$ and 6.1 for $N = 200,000$. This variation leads to the change in the distance R_{jA} between a dipole j and the center of a well-separated cluster A. As a result, the error in the force obtained using the treecode algorithm fluctuates with N as shown in Figure 8 and 9. This suggests, similar to the error bounds for the multipole approximation of Coulomb potential [8] and London dispersion potential [7], the error bound for the multipole approximation of $\phi(\mathbf{x}) = \frac{\sqrt{\pi}}{2}\mathrm{erfc}(|\mathbf{x}|)/|\mathbf{x}|$ is also a decreasing function of R_{jA}.

6. SUMMARY AND CONCLUSIONS

Efficient evaluation of electrostatic interactions is a bottleneck problem in molecular dynamics simulations. Treecode algorithms typically reduce the operation count from $O(N^2)$ to $O(N \log N)$, where N is the number of particles in the system, but there is still much interest in optimizing performance within this class of algorithms. Here we described a treecode algorithm for computing the real space part of the Ewald summation in dipolar systems. The treecode uses multipole approximation in Cartesian coordinates and employs recurrence relations to efficiently compute the necessary Taylor coefficients. We have implemented two different evaluation procedures for applying the approximation recursively. Test results were presented for sets of randomly generated dipoles. The CPU time was consistent with $O(N \log N)$ and the algorithm was effective in controlling the computational error.

The Ewald summation technique poses two distinct computational problems — the evaluation of the real space sum and the reciprocal space sum. The present treecode algorithm evaluates the real space sum in $O(N \log N)$ operations, while the PME method evaluates the reciprocal space sum in $O(N \log N)$ operations. This raises the possibility of combining the two methods to obtain a more efficient hybrid method to reduce the constant of proportionality.

7. ACKNOWLEDGMENTS

This work was supported by a start-up fund from the University of Akron and Michigan Life Sciences Corridor grant #1515. The authors thank Wei-Li Qian for helpful discussions.

8. REFERENCES

[1] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University, New York, 1987.

[2] A. W. Appel. An efficient program for many-body simulation. *SIAM Journal on Scientific and Statistical Computing*, 6(1):85–103, January 1985.

[3] J. Barnes and P. Hut. A hierarchical $o(n \log n)$ force-calculation algorithm. *Nature*, 324(4):446–449, December 1986.

[4] T. Darden, D. York, and L. Pedersen. Particle mesh ewald: An $n \cdot \log(n)$ method for ewald sums in large systems. *Journal of Chemical Physics*, 98(12):10089–10092, June 1993.

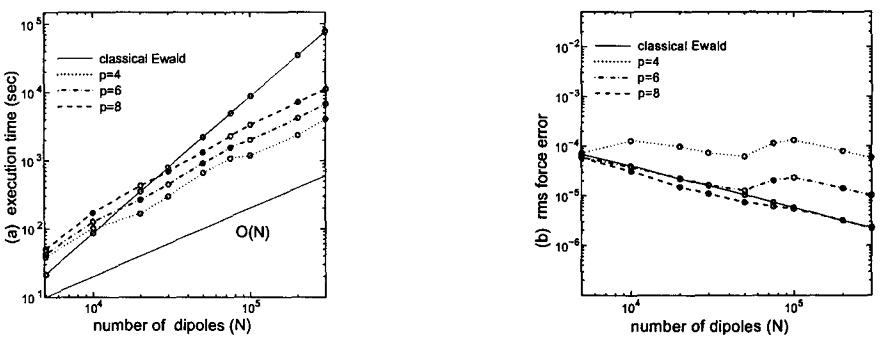

Figure 8: Comparison of classical Ewald method with a neighbor cell list and treecode using pth order expansion and dipole-cluster evaluation procedure. (a) CPU time (force and potential); (b) rms error.

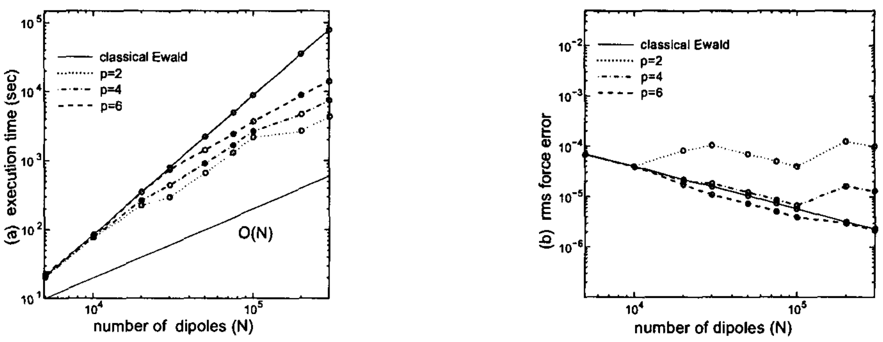

Figure 9: Comparison of the classical Ewald method with a neighbor cell list and treecode using pth order expansion and cluster-cluster evaluation procedure. (a) CPU time (force and potential); (b) rms error.

[5] S. W. de Leeuw, J. W. Perram, and E. R. Smith. Simulation of electrostatic systems in periodic boundary conditions. i. lattice sums and dielectric constants. *Proceedings of the Royal Society of London. Series A*, 373:27–56, 1980.

[6] Z.-H. Duan and R. Krasny. An ewald summation based multipole method. *Journal of Chemical Physics*, 113(9):3492–3495, September 1999.

[7] Z.-H. Duan and R. Krasny. An adaptive treecode for computing nonbonded potential energy in classical molecular systems. *Journal of Computational Chemistry*, 22(2):184–195, December 2000.

[8] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, December 1987.

[9] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the laplace equation in three dimensions. *Acta Numerica*, 229:269, 1997.

[10] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles.* McGraw-Hill, New York, 1981.

[11] R. Kutteh and J. Nicholas. Efficient dipole iteration in polarizable charged systems using the cell multipole method and application to polarizable water. *Computer Physics Communications*, 86(3):227–235, May 1995.

[12] J. W. Perram, H. G. Petersen, and S. W. de Leeuw. An algorithm for the simulation of condensed matter which grows as the 3/2 power of the number of particles. *Molecular Physics*, 65:875–893, 1988.

[13] T. Schlick, R. D. Skeel, A. T. Brunger, L. V. Kale, J. A. Board, Jr., J. Hermans, and K. Schulten. Algorithmic challenges in computational molecular biophysics. *Journal of Computational Physics*, 151(1):9–48, May 1999.

[14] K. E. Schmidt and M. A. Lee. Implementing the fast multipole method in three dimensions. *Journal of Statistical Physics*, 63(5-6):1223–1235, 1991.

[15] W. Smith. Point multipoles in the ewald summation. *CCP5 Newsletter (http://www.dl.ac.uk/CCP/CCP5/ newsletter_index.html)*, (46), October 1998.

[16] A. J. Stone. *The Theory of Intermolecular Forces.* Clarendon Press, Oxford, 1996.

[17] A. Toukmaji, C. Sagui, J. Board, and T. Darden. Efficient particle-mesh ewald based approach to fixed and induced dipolar interactions. *Journal of Chemical Physics*, 113(24):10913–10927, December 2000.

[18] A. Y. Toukmaji and J. J. A. Board. Ewald summation techniques in perspective: a survey. *Computer Physics Communications*, 95(2-3):73–92, June 1996.

[19] Z. Wang and C. Holm. Estimate of the cutoff errors in the ewald summation for dipolar systems. *Journal of Chemical Physics*, 115(14):6351–6359, October 2001.

[20] M. S. Warren and J. K. Salmon. A portable parallel particle program. *Computer Physics Communications*, 87(1-2):266–290, May 1995.

[21] S. Zhang and J. Jin. *Computation of Special Functions.* Wiley, New York, 1996.