

# Efficient implementation of the Barnes-Hut octree algorithm for Monte Carlo simulations of charged systems

GAN ZeCheng & XU ZhenLi\*

*Department of Mathematics, Institute of Natural Sciences,  
and MoE Key Lab of Scientific and Engineering Computing,  
Shanghai Jiao Tong University, Shanghai 200240, China  
Email: ganzecheng1988@sjtu.edu.cn, xuzl@sjtu.edu.cn*

Received April 23, 2013; accepted October 15, 2013; published online February 13, 2014

---

**Abstract** Computer simulation with Monte Carlo is an important tool to investigate the function and equilibrium properties of many biological and soft matter materials solvable in solvents. The appropriate treatment of long-range electrostatic interaction is essential for these charged systems, but remains a challenging problem for large-scale simulations. We develop an efficient Barnes-Hut treecode algorithm for electrostatic evaluation in Monte Carlo simulations of Coulomb many-body systems. The algorithm is based on a divide-and-conquer strategy and fast update of the octree data structure in each trial move through a local adjustment procedure. We test the accuracy of the tree algorithm, and use it to perform computer simulations of electric double layer near a spherical interface. It is shown that the computational cost of the Monte Carlo method with treecode acceleration scales as  $\log N$  in each move. For a typical system with ten thousand particles, by using the new algorithm, the speed has been improved by two orders of magnitude from the direct summation.

**Keywords** electrostatics, Monte Carlo, fast algorithms, octree, colloidal interfaces

**MSC(2010)** 41A58, 82D15, 68P05

---

**Citation:** Gan Z C, Xu Z L. Efficient implementation of the Barnes-Hut octree algorithm for Monte Carlo simulations of charged systems. *Sci China Math*, 2014, 57: 1331–1340, doi: 10.1007/s11425-014-4783-5

---

## 1 Introduction

Monte Carlo methods are often used in computer simulations of coarse-grained biological and soft matter materials in aqueous solution, such as in studying the properties of membranes, DNAs and colloidal suspensions [1, 12]. In these simulations, a bottleneck for high performance computing is to fast and accurately evaluate pairwise electrostatic interactions in the system. For a group of  $N$  charges  $\{q_1e, q_2e, \dots, q_Ne\}$  at  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$ , the total electrostatic energy of the system,  $V$ , can be expressed as,

$$\beta V = \ell_B \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (1.1)$$

where  $\beta$  and  $\ell_B$  are two constants representing the inverse thermal energy and the Bjerrum length of the medium, respectively. A direct summation of this energy scales as  $O(N^2)$  so it becomes prohibitively expensive if large systems are considered.

---

\*Corresponding author

A lot of fast algorithms have been developed to speed up this computation. Due to the long-range nature, simply truncating the infinite system to make a finite simulation volume is never allowed because of strong artifact effect. Periodic boundary conditions could reduce this effect and maintain the particles in the volume but include an infinite series of periodic images. The Ewald summation [10] splits the pairwise interaction into a short-range interaction and a long-range interaction which can be performed in the physical space and in the Fourier space, respectively, and a balance of the computation in the two spaces yields an  $O(N^{3/2})$  algorithm. This can be further simplified to  $O(N \log N)$  when a lattice summation based on fast Fourier transform is utilized, resulting in particle-mesh Ewald [6] and particle-particle particle-mesh Ewald [19] algorithms, popularly employed in molecular dynamics packages. Frequently, it has been interesting to simulate nonperiodic systems. In particular, when a highly charged object is present in the system, periodic boundary conditions have to use very large simulation volume to reduce periodic artifacts. Reaction field models with spherical cavity [34] have been proposed, which include finite number of images to represent the dielectric effects of the bulk solvent due to the cavity. Since the particle systems composed of source and image charges are extremely nonuniform, Barnes-Hut treecode [3,15,30,33] and fast multipole [16,20,22] algorithms will be beneficial to simulate such systems.

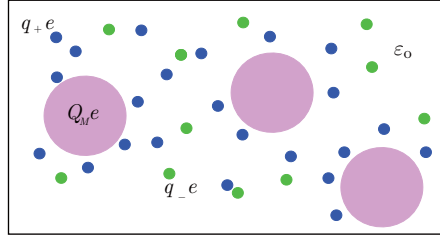
The treecode algorithms originate from Appel [2] and Barnes and Hut [3] in the field of astrophysics. Appel used a binary tree structure in which the clusters can have arbitrary shapes, while Barnes and Hut used an octree data structure where all clusters are cubic boxes. These algorithms divide the particles into a hierarchical tree structure of clusters, and compute the particle-cluster interaction by using the multipole expansion of the Green's function along the tree. The Barnes-Hut method and its generalizations have been often used in practical simulations due to its conceptually simple and easy-to-implement tree construction. Treecode algorithms have an  $O(N \log N)$  computational complexity for pairwise interaction of  $N$  particles and they can be applied to both periodic and non-periodic boundary conditions [8,9,25], and also Yukawa [24] and generalized Born [35] potentials. Additional step by converting them into local expansions leads to the fast multipole algorithms [5,17,38] (see Ying [37] for a tutorial review), which could improve the complexity up to  $O(N)$ .

In Monte Carlo (MC) simulations, the algorithm for electrostatics module should be specially made. Different from the collective motion of all particles in dynamic simulations, one-particle displacement is preferred in each MC step in order to gain a better sampling efficiency [12]. In this displacement strategy, only particle interactions related to the chosen particle are required to calculate. Suppose the  $i$ -th particle is chosen, then the energy is,  $\beta V_i = \ell_B \sum q_i q_j / r_{ij}$ , a summation over index  $j \neq i$ . This expression is calculated twice to compare the energy difference before and after the displacement of particle  $i$ , which is used to decide if the new configuration is accepted or rejected. Obviously, a simple direct evaluation costs  $O(N)$  complexity, and the cost of using Ewald summation will be  $O(N^{1/2})$ . The lattice summations are not advantaged in one-particle-displacement MC simulations. This motivates the present work to develop an  $O(\log N)$  calculation for each MC displacement with the Barnes-Hut-type algorithm targeting at large scale simulations of nonuniformly distributed charged systems.

Our algorithm for constructing an efficient electrostatic module of MC simulations is based on the following intuitive observation. For a given octree structure, the operations of calculating  $V_i$  in each MC step will be  $O(\log N)$  if the multipole expansion is applied, and therefore we should also make a local modification of the tree structure within  $O(\log N)$  in every particle displacement. This can be guaranteed if the local adjustment of the octree data is only done for clusters related to the chosen particle. By this strategy, we test the performance of the treecode algorithm in simulating a typical charged system in colloidal suspensions, which demonstrates its attractive feature for a wider application in related complex biological or soft matter systems.

## 2 Primitive model of charged systems and Monte Carlo simulations

Let us start with a typical charged system for colloidal suspensions composed of macroions and small ions, schematically shown in Figure 1. One common concern in different disciplines is the ionic structure



**Figure 1** Schematic illustration of the primitive model of electrolytes, including spherical macroions (pink) with valence  $Q_M e$ , counterions (blue) and coions (green) with valence  $q_+$  and  $q_-$ , respectively. These particles at different scales are mobile in a solvent which affects the ions by its dielectric permittivity  $\varepsilon_0$

near a charged surface, which is the so-called electric double layer (EDL) [11], since in the EDL ions are strongly correlated and ion distribution cannot be correctly predicted by mean-field theory. The primitive model is often used to describe EDLs [26,32], because it takes into account ion-ion interactions explicitly. Within the primitive model framework, ionic species are represented by charged hard spheres of different sizes with charges at spherical centers. The solvent is treated implicitly and only enters the model with its relative permittivity  $\varepsilon_0$ . The Hamiltonian of an EDL system can be written as a pairwise summation over all macro- and small ions,  $V = \sum_{i < j} V_{ij}$ , and  $V_{ij}$  is the superposition of the electrostatic interaction and the hard-sphere repulsion between two particles,

$$\beta V_{ij} = \begin{cases} \frac{\ell_B q_i q_j}{r_{ij}}, & \text{if } r_{ij} > (\tau_i + \tau_j)/2, \\ \infty, & \text{if } r_{ij} \leq (\tau_i + \tau_j)/2, \end{cases} \quad (2.1)$$

where  $\beta = 1/k_B T$  is the inverse thermal energy,  $\ell_B = e^2/4\pi\varepsilon_0\varepsilon_0 k_B T$  is the Bjerrum length,  $\varepsilon_0$  is the vacuum permittivity and  $\varepsilon_0$  is the relative permittivity,  $q_i$  is the valence,  $\tau_i$  is the diameter of ion, and  $r_{ij}$  is the distance between two ions.

In equilibrium, the system follows the Boltzmann distribution  $P(X) = e^{-\beta V(X)}/Z$ , where  $X$  is a sample in a high dimensional space ( $3N$  dimensions where  $N$  is the number of particles) and presents a possible configuration of the system, and  $Z = \int_{\Omega} e^{-\beta V(X)} dX$  is a normalization constant. The partition function  $Z$  is an integral over the whole sample space  $\Omega$ . The Monte Carlo (MC) algorithm [12] is a computational approach to obtain equilibrium distribution of such a system. It samples  $M$  configurations in  $\Omega$  by a Markov chain,  $X_1, X_2, \dots, X_M$ , such that for any configuration  $X$  the following limit is satisfied,

$$\lim_{M \rightarrow \infty} \frac{M_X}{M} = P(X), \quad (2.2)$$

where  $M_X$  is the number of configurations  $X$  out of the sampled  $M$  configurations.

In general, the Markov chain is constructed through the Metropolis principle [31]. The implementation of the Metropolis MC algorithm can be divided into the following steps:

1. Choose an initial configuration  $X_1$  from the sample space. This can be done by randomly placing particles in simulation volume. Set index  $m = 1$ .
2. Choose a trial configuration  $X_t$ , compute the ratio  $\Theta = P(X_t)/P(X_m) = e^{-\beta[V(X_t) - V(X_m)]}$ , and generate a random number  $\alpha$  from  $[0, 1]$ . Set the next configuration of the Markov chain,  $X_{m+1} = X_t$  if  $\alpha \leq \Theta$ , otherwise copy the old configuration as the new configuration  $X_{m+1} = X_m$ .
3. Let index  $m = m + 1$  and go to Step 2 for next configuration.

With the Markov chain, any equilibrium property of the system,  $\langle B \rangle$ , which is an ensemble average, can be calculated through,

$$\langle B \rangle \approx \frac{1}{M} \sum_X M_X B(X). \quad (2.3)$$

Here,  $B(X)$  is the value of  $B$  for a state  $X$ .

The benefit of the Metropolis algorithm is to avoid computing  $Z$  directly, and instead that only energy difference of two configurations  $V(X_t) - V(X_m)$  is needed to compute, which becomes the most expensive part of the algorithm. Another benefit, which can be seen from Step 2, is that the algorithm allows a flexible choice of a candidate configuration which should satisfy the detailed balance condition,  $P(X)p(X \rightarrow Y) = P(Y)p(Y \rightarrow X)$ , where  $p$  is the transition probability. A suitable choice of the candidate configuration will greatly increase the sampling efficiency. In consideration of computing the energy difference in a low price, Metropolis MC algorithm usually displaces one randomly chosen particle to generate a new configuration in each step [12], which has similar sampling efficiency as displacing multiple particles for a given acceptance ratio, but greatly improves the computation speed of the electrostatic energy.

### 3 Monte Carlo implementation of the tree algorithm

#### 3.1 Multipole expansion and treecode algorithm

In this section, we overview the mathematical formulation of the treecode algorithm which will be used for energy calculation, i.e., the multipole approximation for the particle-cluster interaction with truncating multivariate Taylor and the recurrence relations for the expansion coefficients. In Cartesian coordinates,  $\mathbf{r} = (x_1, x_2, x_3)$ , the multivariate Taylor expansion of the Green's function  $f(\mathbf{r}) = 1/|\mathbf{r}|$  at  $\mathbf{r}'$  can be written as,

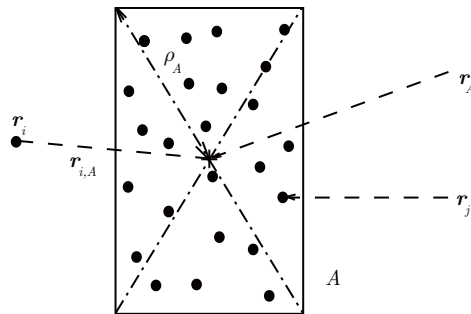
$$f(\mathbf{r}) = \sum_{\|\mathbf{n}\|=0}^{\infty} \frac{1}{\mathbf{n}!} D_{\mathbf{r}}^{\mathbf{n}} f(\mathbf{r}') (\mathbf{r} - \mathbf{r}')^{\mathbf{n}}, \quad (3.1)$$

where the multi-index notation  $\|\mathbf{n}\| = n_1 + n_2 + n_3$ ,  $\mathbf{n}! = n_1!n_2!n_3!$ ,  $D_{\mathbf{r}}^{\mathbf{n}} = \partial^{\|\mathbf{n}\|} / (\partial x_1^{n_1} \partial x_2^{n_2} \partial x_3^{n_3})$ , and  $(\mathbf{r} - \mathbf{r}')^{\mathbf{n}} = (x_1 - x_1')^{n_1} (x_2 - x_2')^{n_2} (x_3 - x_3')^{n_3}$ . We will use this expression for the multipole approximation of a particle  $i$  and particles in a distant cluster  $A$  which has a center  $\mathbf{r}_A$  (see Figure 2). The electrostatic energy between particle  $i$  and particles in the cluster  $A$  is,

$$\beta V_{i,A} = \ell_B \sum_{j \in A} \frac{q_i q_j}{|\mathbf{r}_j - \mathbf{r}_i|}. \quad (3.2)$$

Substituting (3.1) into (3.2) by setting  $\mathbf{r} = \mathbf{r}_j - \mathbf{r}_i$  and  $\mathbf{r}' = \mathbf{r}_A - \mathbf{r}_i$ , and truncating the expansion at  $\|\mathbf{n}\| = p$ , we obtain the  $p$ -th order multipole approximation to  $V_{i,A}$  as,

$$\begin{aligned} \beta V_{i,A} &= \ell_B \sum_{j \in A} q_i q_j f(\mathbf{r}_j - \mathbf{r}_i) \\ &\approx \ell_B q_i \sum_{\|\mathbf{n}\|=0}^p \frac{1}{\mathbf{n}!} D_{\mathbf{r}}^{\mathbf{n}} f(\mathbf{r}_A - \mathbf{r}_i) \sum_{j \in A} q_j (\mathbf{r}_j - \mathbf{r}_i)^{\mathbf{n}} \\ &= \ell_B q_i \sum_{\|\mathbf{n}\|=0}^p T_{\mathbf{n}}(\mathbf{r}_A - \mathbf{r}_i) M_A^{\mathbf{n}}, \end{aligned} \quad (3.3)$$



**Figure 2** Illustration of interaction between a particle  $i$  and particles in a distant cluster  $A = \{\mathbf{r}_j, j = 1, 2, \dots, N_A\}$ .  $\mathbf{r}_{i,A}$  is the displacement between the particle to the cluster center,  $\mathbf{r}_A$  is the center of the cluster, and  $\rho_A$  is the radius of the cluster

where  $T_{\mathbf{n}}(\mathbf{r}_A - \mathbf{r}_i) = \frac{1}{\mathbf{n}!} D_{\mathbf{r}}^{\mathbf{n}} f(\mathbf{r}_A - \mathbf{r}_i)$  is the  $\mathbf{n}$ -th order coefficient of the multidimensional Taylor expansion and  $M_A^{\mathbf{n}} = \sum_{j \in A} q_j (\mathbf{r}_j - \mathbf{r}_A)^{\mathbf{n}}$  is the corresponding multipole moment of cluster  $A$ . It should be noted that the moment  $M_A^{\mathbf{n}}$  is independent of the distant particle  $i$ , and thus it can be pre-calculated once for the use of alternating the distant particle  $i$ . Another important observation is that the coefficient  $T_{\mathbf{n}}(\mathbf{r}_A - \mathbf{r}_i)$  only depends on the displacement between the particle  $i$  and the cluster center and is independent of particles in the cluster, and the calculation of this quantity is supposed to be efficient in simulations.

The Taylor coefficients  $T_{\mathbf{n}}$  can be calculated by using recurrence relations [9],

$$\|\mathbf{n}\| r^2 T_{\mathbf{n}}(\mathbf{r}) + (2\|\mathbf{n}\| - 1) \sum_{i=1}^3 x_i T_{\mathbf{n} - \mathbf{e}_i}(\mathbf{r}) + (\|\mathbf{n}\| - 1) \sum_{i=1}^3 T_{\mathbf{n} - 2\mathbf{e}_i}(\mathbf{r}) = 0, \quad (3.4)$$

where  $r = |\mathbf{r}|$ . Since the Taylor coefficient  $T_{\mathbf{n}}(\mathbf{r}) = 0$  when a certain index  $n_i$  is negative, these recurrence relations allow us to efficiently calculate expansion coefficients in the multipole approximation. The error of the  $p$ -th order multipole approximation between particle  $i$  and a particle at  $\mathbf{r}_j$  inside cluster  $A$  can be estimated by bounding the sum of the  $n > p$  terms, which is given by [9],

$$\sum_{n=p+1}^{\infty} \sum_{\|\mathbf{n}\|=n} |T_{\mathbf{n}}(\mathbf{r}_A - \mathbf{r}_i)(\mathbf{r}_j - \mathbf{r}_A)^{\mathbf{n}}| \leq \frac{1}{R} \left( \frac{\gamma^{p+1}}{1 - \gamma} \right), \quad (3.5)$$

where  $\rho_A$  is the radius of the cluster equal to half length of the box diagonal,  $R = r_{i,A}$  and  $\gamma = \rho_A/R$  should be less than 1. We can see increasing either  $R$  or the expansion order  $p$  could improve the accuracy of the multipole expansion. In practice, the error could be even much smaller due to the error cancellation of different particles in  $A$ , in particular, when the particle distribution is uniform.

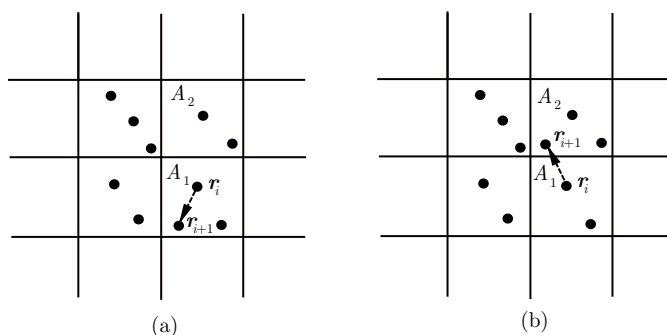
### 3.2 Calculation of electrostatic energy

The calculation of electrostatic energy in the Hamiltonian of charged systems is based on a hierarchical octree of the data structure. Each node in the tree is a cluster of particles. One of them is the root cluster, which is a cubic box enclosing the whole simulation domain (usually the smallest box including all the particles is taken), and we say it is the zeroth level of the tree structure. In our algorithm, the root cluster is bisected in each coordinate direction to yield eight children clusters which form the first level of the tree. This process is applied recursively on the children clusters until a user-specified level  $L_{\max}$  is reached. The clusters at level  $L_{\max}$  are also called leaves, and  $L_{\max}$  logarithmically depends on the number of particles in the system.

The generation of the tree structure is order  $N \log N$  algorithm. Once the tree has been constructed, it will remain a fixed structure throughout the MC simulation and only a local modification is allowed. In the data structure, each node, i.e., a certain cluster  $A$ , includes the following information of  $A$ :

1. The number of particles in cluster  $A$ ,  $N_A$ . For example, the root cluster has  $N$  particles.
2. The radius of the cluster  $\rho_A$ , which is equal to a half of the box diagonal length.
3. The minimal, middle and maximal values of the  $x, y, z$  coordinates of the cluster box.
4. A value,  $l_A$ , to indicate at which level the cluster is located.
5. An index array which stores all particles in the cluster.
6. All the moments for the truncated multipole expansion.
7. The pointers to its eight children if  $A$  is not a leaf cluster.

The potential energy of a particle  $i$  interacting with other particles is the summation of particle-cluster interactions by the multipole approximation or particle-particle interactions by the direct summation. It is calculated by traversing the tree recursively, starting from the root cluster. If the particle-cluster interaction being employed is determined according to the multipole acceptance criterion (MAC),  $\rho_A/r_{i,A} \leq \theta$ , and if  $N_A \geq N_0$  is satisfied, where  $\theta \in [0, 1]$  and the minimum expansion number  $N_0$  are two user-specified parameters. If both two conditions are satisfied, the multipole approximation is used. If the MAC is not satisfied, but  $N_A > N_0$  and  $A$  is not a leaf cluster, then the eight children of  $A$  will be examined, because the radius of the children clusters will be smaller, and the MAC is more likely to be satisfied. For all the other cases, the direct summation method is used.



**Figure 3** Two possible cases which could happen for an MC trial move. At the same level, (a) the particle  $i$  stays inside a cluster  $A_1$ ; (b) it moves from  $A_1$  to another cluster  $A_2$

### 3.3 Local modification of the tree structure

Due to the possible position change of the chosen particle, a minor modification to the data information of the tree structure should be done in each MC displacement. In order to guarantee that the local adjustment procedure can be accomplished in  $O(\log N)$  operations, we need to use the tracking information for all the particles, which is stored initially. At a certain level we know the cluster including this particle, the tracking information of this particle essentially remembers the location at the index array of this particle. This information is useful in reducing the operations.

Suppose a trial move takes place during an MC step of trying to move the  $i$ th particle. At each level of the octree, there could be two possible cases (see Figure 3): (a) Particle  $i$  stays inside the original cluster  $A_1$ , or (b) it moves from  $A_1$  to another cluster  $A_2$ . Suppose Case (a) happens, then we only need to update the moments stored in the node of  $A_1$ . If it is Case (b), then we need an “add” operation to  $A_2$  and a “delete” operation to  $A_1$ . In the add operation, we need to update  $N_{A_2}$ , the moments by adding the contribution from  $i$ , and the index array adding the index  $i$  at the end of this index array, and finally the tracking information of  $i$ . In the delete operation, we update  $N_{A_1}$  and the moments by extracting the contribution from  $i$ , and in the index array, we replace  $i$ -th index by the one at the end of the array and update the tracking information of this index. Both the add and delete operations are performed recursively.

The new MC algorithm with treecode acceleration is summarized as follows: Initially, the locations and charges of all particles are generated, and the parameters such as the MAC parameter  $\theta$ , the order of the multipole expansion  $p$ , and the maximal expansion number  $N_0$  are prescribed. Then the octree structure and the tracking information for each particle are built. At every MC step, we calculate the electrostatic energy with  $O(N)$  operations. The cell-list algorithm is used for the short range interaction between particles, which has  $O(1)$  operations. The local adjustment procedure is then performed to update the data information in the tree. If the trial move is rejected, the tree structure remains unchanged.

## 4 Numerical results

In order to test the performance of the algorithm, we have implemented the Metropolis MC simulation to calculate the equilibrium structure of the spherical EDLs. The EDL is composed of positively and negatively charged ions surrounding a macroion, all ions are of spherical shapes. The spherical cell model [26] is used to define the simulation volume, which confines all ions in a spherical cell, i.e., if any ion moves across the cell boundary in an MC step, the trial move will be rejected by the hard boundary. The macroion is fixed and concentric with the spherical cell, while all the small ions can move freely inside the cell. Due to its importance in colloidal science, biophysics, and electrochemical energy and so on, computer simulations of the EDL structure have attracted a wide interest [7, 21, 27–29].

In the calculations, we take the Bjerrum length  $\ell_B = 7.14\text{\AA}$ , which represents the water solvent at room temperature. Besides the macroion, only counterions are present in each system, which have a uniform diameter  $\tau = 4\text{\AA}$  and positive charge  $q = 2$ . The radius of the macroion takes  $a = 30\text{\AA}$ , which is

at the same scale as a charged surfactant micelle [26]. The radius of the simulation cell is  $R_{\text{cell}} = 80\text{\AA}$ . In testing the timing, the number of counterions,  $N$ , is varied from 10 to 10k, the valence of the macroion takes  $Q_M = -qN$  to guarantee the electroneutrality of each system.

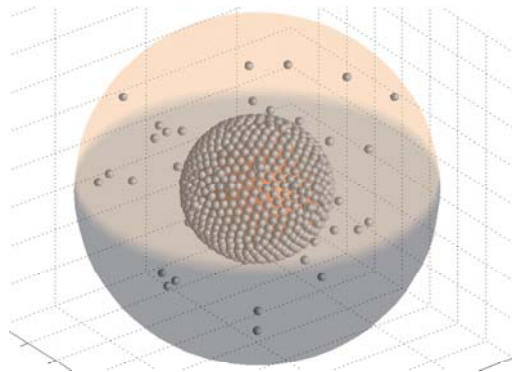
In electrostatic algorithm based on the Barnes-Hut tree structure, the MAC parameter takes two values of  $\theta = 0.5$  and  $0.3$  for comparison, and the order of the multipole expansion  $p$  varies between 0 and 10. The minimal expansion number  $N_0$  is optimized as a function of  $p$  by pre-computing execution times, which is determined to be the smallest number of particles in a cluster such that the multipole approximation is more efficient than the direct summation. In each simulation,  $10^6 N$  MC trial moves are attempted for the convergence of the empirical distribution. The computations were performed on a Linux machine with 2.67 GHz CPU and 48 G memory.

In order to verify the accuracy of the algorithm, we randomly choose a snapshot configuration of the spherical EDL with  $N = 10$  to 10k, and calculate the error of different orders based on this configuration. A typical configuration with  $N = 800$  is illustrated in Figure 4, where we can see the charge distribution is not uniform and most mobile ions are aggregated near the surface. Although the strong surface charge is not physical, this acts as a good test for the performance of the multipole expansion since there is no error cancellation for charges in a cluster having the same sign. We measure the relative error  $E$  of the treecode algorithm, which is defined by,

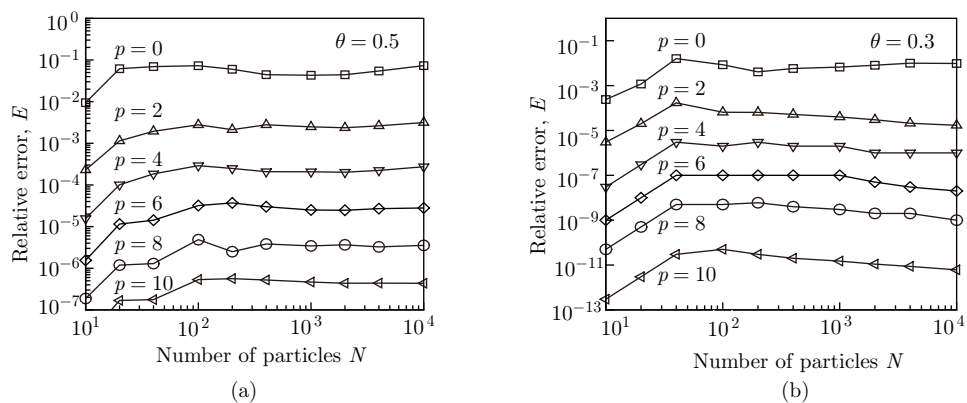
$$E = \left( \frac{\sum_{i=1}^N |V_i - \bar{V}_i|^2}{\sum_{i=1}^N |V_i|^2} \right)^{1/2}, \quad (4.1)$$

where  $V_i$  is the exact potential on particle  $i$  obtained by the direct summation method and  $\bar{V}_i$  is the potential obtained using treecode algorithm. The results are shown in Figure 5. We can see that when  $N \geq 100$ , the error is almost independent of the system size  $N$ . The relative error decreases with the increase of order  $p$ . Although most counterions assemble near the macroion surface and the system is not uniform, when  $p = 2$  the error is already less than 1% for  $\theta = 0.5$  and 0.1% for  $\theta = 0.3$ .

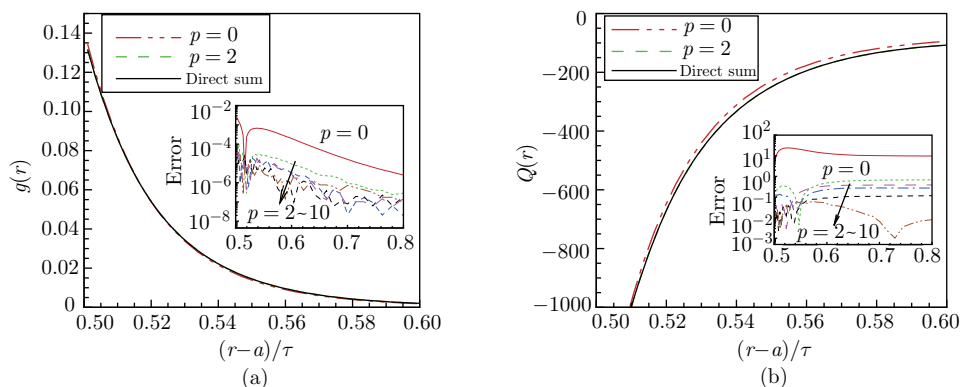
Using Monte Carlo simulations, we also test the accuracy by investigating two physical properties of double layer structure: The radial distribution function (RDF),  $g(r)$ , of the counterions and the integrated charge distribution function (ICDF),  $Q(r)$ , of the systems. In simulations, it is calculated using the average particle in a bin  $N(r, r + \Delta r)$  divided the volume of the bin and normalized the sum of all bins. The RDF describes the normalized average density along the radial direction. And the ICDF  $Q(r)$  is a function of  $r$ , which represents the average charge within the spherical region of radius  $r$ . Clearly, we have  $Q(a) = Q_M$  and  $Q(R_{\text{cell}}) = 0$ . To observe the accuracy of these two functions, we plot the simulation results of  $N = 800$  and  $\theta = 0.5$  in Figure 6. We see both functions converge to the results of direct summation, starting from  $p = 2$ . Shown in the inset plots, the errors are substantially in the same order as  $p \geq 2$  due to the fluctuation of the MC sampling, demonstrating that the multipole expansion with  $p = 2$  provides enough accuracy for describing the physical properties of the studied system. It



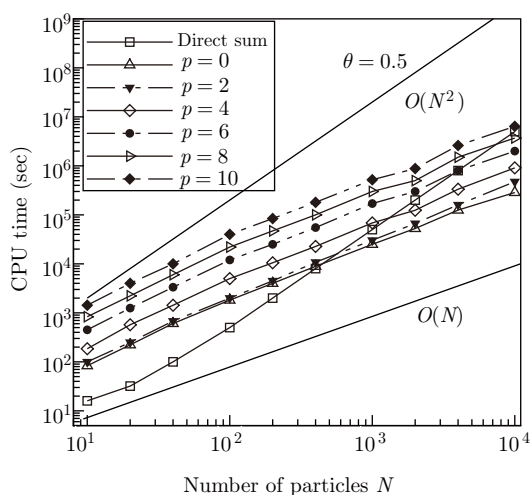
**Figure 4** A snapshot configuration of charge distribution around a macroion with 800 counterions



**Figure 5** Relative errors in electrostatic potential energy,  $E$ , computed by the treecode algorithm with multipole approximation order  $p$  from 0 to 10. (a)  $\theta = 0.5$ ; and (b)  $\theta = 0.3$



**Figure 6** Radial distribution functions (a) and integrated charge distribution functions (b) of a typical system with  $N = 800$ , computed by using direct summation, and multipole approximations with  $\theta = 0.5$  and order  $p$  from 0 to 10. The curves of  $p = 2 \sim 10$  and the direct summation are overlapping, and the errors deviated from the direct summation can be found from the inset plots



**Figure 7** CPU times in MC simulations of the spherical EDLs with the direct summation and treecode acceleration for  $p = 0 \sim 10$  as a function of particle number  $N$



can be observed that both  $g(r)$  and  $Q(r)$  exponentially decay near the charged interface, in agreement with the prediction from the strong-coupling theory [4]. In contrast, the decay rate predicted by the classical Poisson-Boltzmann theory is algebraic [4], and thus these simulation results demonstrate that the classical theory fails for strong-coupling charged systems [18, 23].

Finally, the CPU times of the MC algorithm are compared for different strategies of the electrostatic calculations, with different system size  $N$  and expansion order  $p$ . The results have been shown in Figure 7. Since the number of MC steps is of  $O(N)$ , it is illustrated that the CPU times for those curves with treecode acceleration scale as  $O(N \log N)$  while that of direct summation scales as  $O(N^2)$ . The case of  $p = 2$ , which is accurate enough to provide correct RDF and ICDF curves, has the break-even point about  $N = 600$ . Using the treecode acceleration, we find much larger systems can be explored, and the simulation for a system of  $N = 10\text{k}$  can be done in a few days using the new MC algorithm, which is around 100 times faster than that of the direct summation.

## 5 Conclusions

In this paper, we have developed an efficient generalization of the Barnes-Hut octree algorithm in Monte Carlo simulations of charged systems. In order to adapt the one-particle displacement of Metropolis Monte Carlo simulations, the algorithm introduces a local modification strategy of the tree structure, which leads to a low computational cost, namely,  $O(\log N)$  complexity in each Monte Carlo step. This allows us to simulate equilibrium properties of a system with more than ten thousand charged particles in personal computers.

Future goals of this project includes the parallelization of the treecode algorithm [14, 33], the incorporation into the multiscale reaction-field modeling of electrolytes and with image charges for colloidal suspensions [13, 36], and the exploration of properties of larger scaled systems.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (Grant Nos. 11101276 and 91130012). The second author acknowledged the support from the Alexander von Humboldt Foundation for a research stay at the Institute of Computational Physics, University of Stuttgart.

## References

- 1 Allen M P, Tildesley D J. *Computer Simulations of Liquids*. Oxford: Oxford University Press, 1987
- 2 Appel A. An efficient program for many-body simulations. *SIAM J Sci Stat Comput*, 1985, 6: 85–103
- 3 Barnes J, Hut P. A hierarchical  $O(N \log N)$  force-calculation algorithm. *Nature*, 1986, 324: 446–449
- 4 Boroudjerdi H, Kim Y W, Naji A, et al. Statics and dynamics of strongly charged soft matter. *Phys Rep*, 2005, 416: 129–199
- 5 Cheng H, Greengard L, Rokhlin V. A fast adaptive multipole algorithm in three dimensions. *J Comput Phys*, 1999, 155: 468–498
- 6 Darden T A, York D M, Pedersen L G. Particle mesh Ewald: An  $N \log(N)$  method for Ewald sums in large systems. *J Chem Phys*, 1993, 98: 10089–10092
- 7 Deserno M, Jiménez-ángel F, Holm C, et al. Overcharging of dna in the presence of salt: Theory and simulation. *J Phys Chem B*, 2001, 105: 10983–10991
- 8 Duan Z H, Krasny R. An Ewald summation based multipole method. *J Chem Phys*, 2000, 113: 3492–3495
- 9 Duan Z H, Krasny R. An adaptive treecode for computing nonbonded potential energy in classical molecular systems. *J Comput Chem*, 2001, 22: 184–195
- 10 Ewald P P. Die berechnung optischer und elektrostatischer gitterpotentiale. *Ann Phys*, 1921, 369: 253–287
- 11 French R H, Parsegian V A, Podgornik R, et al. Long range interactions in nanoscale science. *Rev Mod Phys*, 2010, 82: 1887–1944
- 12 Frenkel D, Smit B. *Understanding Molecular Simulation: From Algorithms to Applications*. New York: Academic Press, 2002
- 13 Gan Z, Xu Z. Multiple-image treatment of induced charges in Monte Carlo simulations of electrolytes near a spherical dielectric interface. *Phys Rev E*, 2011, 84: 016705

- 14 Geng W, Krasny R. A treecode-accelerated boundary integral Poisson-Boltzmann solver for electrostatics of solvated biomolecules. *J Comput Phys*, 2013, 247: 62–78
- 15 Gibbon P, Speck R, Karmakar A, et al. Progress in mesh-free plasma simulation with parallel tree codes. *Plasma Sci IEEE Trans*, 2010, 38: 2367–2376
- 16 Greengard L, Rokhlin V. A fast algorithm for particle simulations. *J Comput Phys*, 1987, 73: 325–348
- 17 Greengard L, Rokhlin V. A new version of the Fast Multipole Method for the Laplace equation in three dimensions. *Acta Numer*, 1997, 6: 229–269
- 18 Grosberg A Y, Nguyen T T, Shklovskii B I. Colloquium: The physics of charge inversion in chemical and biological systems. *Rev Mod Phys*, 2002, 74: 329–345
- 19 Hockney R W, Eastwood J W. *Computer Simulation Using Particles*. Boca Raton, FL: CRC Press, 1988
- 20 Kabadshow I, Dachsel H. The error-controlled fast multipole method for open and periodic boundary conditions. In: *Fast Methods for Long-Range Interactions in Complex Systems*, vol. 6. Jülich: Forschungszentrum Jülich, 2011, 85–114
- 21 Kondrat S, Georgi N, Fedorov M V, et al. A superionic state in nano-porous double-layer capacitors: insights from monte carlo simulations. *Phys Chem Chem Phys*, 2011, 13: 11359–11366
- 22 Lashuk I, Chandramowlishwaran A, Langston H, et al. A massively parallel adaptive fast-multipole method on heterogeneous architectures. *Commun ACM*, 2012, 55: 101–109
- 23 Lau A W C, Lukatsky D B, Pincus P, et al. Charge fluctuations and counterion condensation. *Phys Rev E*, 2002, 65: 051502
- 24 Li P, Johnston H, Krasny R. A Cartesian treecode for screened Coulomb interactions. *J Comput Phys*, 2009, 228: 3858–3868
- 25 Lindsay K, Krasny R. A particle method and adaptive treecode for vortex sheet motion in 3-D flow. *J Comput Phys*, 2001, 172: 879–907
- 26 Linse P. Simulation of charged colloids in solution. *Adv Polym Sci*, 2005, 185: 111–162
- 27 Linse P. Electrostatics in the presence of spherical dielectric discontinuities. *J Chem Phys*, 2008, 128: 214505
- 28 Lyubartsev A P, Tang J X, Janmey P A, et al. Electrostatically induced polyelectrolyte association of rodlike virus particles. *Phys Rev Lett*, 1998, 81: 5465–5468
- 29 Manzanares J, Murphy W, Mafe S, et al. Numerical simulation of the nonequilibrium diffuse double layer in ion-exchange membranes. *J Phys Chem*, 1993, 97: 8524–8530
- 30 Marzouk Y M, Ghoniem A F. *K*-means clustering for optimal partitioning and dynamic load balancing of parallel hierarchical *n*-body simulations. *J Comp Phys*, 2005, 207: 493–528
- 31 Metropolis N, Rosenbluth A W, Rosenbluth M N, et al. Equation of state calculations by fast computing machines. *J Chem Phys*, 1953, 21: 1087–1092
- 32 Walker D A, Kowalczyk B, de la Cruz M O, et al. Electrostatics at the nanoscale. *Nanoscale*, 2011, 3: 1316–1344
- 33 Winkel M, Speck R, Hübner H, et al. A massively parallel, multi-disciplinary Barnes-Hut tree code for extreme-scale *N*-body simulations. *Comput Phys Comm*, 2012, 183: 880–889
- 34 Xu Z, Cai W. Fast analytical methods for macroscopic electrostatic models in biomolecular simulations. *SIAM Rev*, 2011, 53: 683–720
- 35 Xu Z, Cheng X, Yang H. Treecode-based generalized Born method. *J Chem Phys*, 2011, 134: 064107
- 36 Xu Z, Liang Y, Xing X. Mellin transform and image charge method for dielectric sphere in an electrolyte. *SIAM J Appl Math*, 2013, 7: 1396–1415
- 37 Ying L. A pedestrian introduction to fast multipole methods. *Sci China Math*, 2012, 55: 1043–1051
- 38 Ying L, Biros G, Zorin D. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J Comput Phys*, 2004, 196: 591–626