JCTC Journal of Chemical Theory and Computation

Article

Cite This: J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

pubs.acs.org/JCTC

Periodic Coulomb Tree Method: An Alternative to Parallel Particle Mesh Ewald

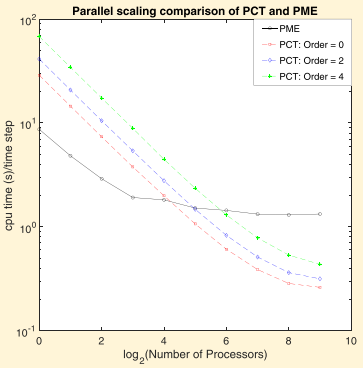
Henry A. Boateng*,†,‡

†Department of Mathematics, San Francisco State University, 1600 Holloway Ave., San Francisco, California 94132, United States

‡Department of Mathematics, Bates College, 2 Andrews Rd., Lewiston, Maine 04240, United States

Supporting Information

ABSTRACT: Particle mesh Ewald (PME) is generally the method of choice for handling electrostatics in simulations with periodic boundary conditions. The excellent efficiency of PME on low processor counts is largely due to the use of the fast Fourier transform (FFT). However, due to the FFT's high communication cost, PME scales poorly in parallel. We develop a periodic Coulomb tree (PCT) method for electrostatic interactions in periodic boundary conditions as an alternative to PME in parallel simulations. We verify the accuracy of PCT by comparison of structural and dynamical properties of three different systems obtained via MD simulations using PME and PCT and provide parallel timing comparisons of the two methods on up to 1024 cores.



1. INTRODUCTION

Periodic boundary conditions are employed in a significant number of molecular simulations of bulk properties to reduce surface effects.[1] Consider a system of N interacting point charges $q_1, q_2, ..., q_N$ at positions $r_1, r_2, ..., r_N$ in a box of length L, with periodic boundary conditions and with the additional constraint that $\sum_{i=1}^{N} q_i = 0$. Let the set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ be a basis for the simulation cell, that is, $\mathbf{r}_i = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3$ for $i = 1...N$ and c_j is a real number for $j = 1...3$. The volume of the simulation cell $V = |\mathbf{v}_1 \cdot \mathbf{v}_2 \times \mathbf{v}_3|$. We also define a basis for the reciprocal space as $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ with $\mathbf{w}_1 = 2\pi\frac{\mathbf{v}_2 \times \mathbf{v}_3}{\mathbf{v}_1 \cdot \mathbf{v}_2 \times \mathbf{v}_3}$, $\mathbf{w}_2 = 2\pi\frac{\mathbf{v}_3 \times \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_2 \times \mathbf{v}_3}$, and $\mathbf{w}_3 = 2\pi\frac{\mathbf{v}_1 \times \mathbf{v}_2}{\mathbf{v}_1 \cdot \mathbf{v}_2 \times \mathbf{v}_3}$. The Coulomb energy of the system can be written as

$$U(r_1, ..., r_N) = \frac{1}{2}\sum_{i,j=1}^{N}{}' \sum_{\mathbf{n}} \frac{q_i q_j}{|\mathbf{r_i} - \mathbf{r_j} + \mathbf{n}L|} \tag{1}$$

where $\mathbf{n} = (n_1, n_2, n_3)$ identifies a unique cell and n_i is an integer. For example, $\mathbf{n} = (0, 0, 0)$ refers to the fundamental cell or our system of interest. The prime on the sum over $\mathbf{n}$ means that the $i = j$ term is omitted when $\mathbf{n} = (0, 0, 0)$.

Ewald[1−3] recast equation 1 into absolutely convergent series, one in real space dominated by the sum

$$U_{\text{dir}} = \frac{1}{2}\sum_{i<j}^{N^*}{}' \sum_{\mathbf{n}} \frac{q_i q_j \text{erfc}(\alpha|\mathbf{r_i} - \mathbf{r_j} + \mathbf{n}L|)}{|\mathbf{r_i} - \mathbf{r_j} + \mathbf{n}L|} \tag{2}$$

and the other in reciprocal space given by

$$U_{\text{rec}} = \frac{1}{2V}\sum_{\mathbf{k}\neq 0} \frac{\exp(-k^2/4\alpha^2)}{k^2}|S(\mathbf{k})|^2 \tag{3}$$

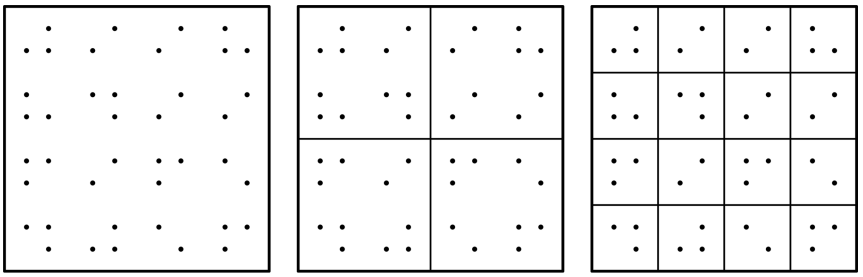
where $\mathbf{k} = k_1\mathbf{w}_1 + k_2\mathbf{w}_2 + k_3\mathbf{w}_3$, k_i is an integer for $i = 1...3$, $k = |\mathbf{k}|$, and

$$S(\mathbf{k}) = \sum_{i=1}^{N} q_i \exp(i\mathbf{k}\cdot\mathbf{r_i}) \tag{4}$$

The decay rate of the sums in eqs 2 and 3 is controlled by α. For small values of α, $\text{erfc}(\alpha r)$ decays slowly, but correspondingly, $\exp(-k^2/4\alpha^2)$ decays quickly. The reverse is seen for large α values. With a careful choice of parameters, the conventional Ewald sum for a system can be evaluated at $O(N^{3/2})$ cost.[3,4] Another approach to computing the conventional Ewald sum is via a Fourier Poisson approach,[5] which scales as $O(N\log N)$. Particle mesh Ewald (PME)[6,7] is the method employed in a majority of simulations to compute eq 1. PME chooses a large α value to ensure that U_{dir} decays quickly while U_{rec} decays slowly. Then U_{dir} is computed in $O(N)$ time using a cutoff and U_{rec} is evaluated in $O(N\log N)$ by approximating the structure factor $S(\mathbf{k})$ on a mesh using an FFT. Thus, the overall cost for the PME method is $O(N\log N)$.PME has been shown to be very efficient for simulations on a small number of processors.[6−9] However, this efficiency degrades severely with increasing processor count primarily due to the intensive global communications cost for the parallel 3D FFT. There are ongoing efforts to develop parallel 3D FFT algorithms, which scale to large processors using 2D decompositions.[10−13] However, these algorithms are

Received: June 28, 2019
Published: November 21, 2019

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX



(a) Root: Level 0 (b) Level 1 (c) Level 2

Figure 1. 2D hierarchical clustering with $N_0 = 3$.

typically developed for specific hardwares, and they lose performance when ported to different systems with different network topologies.[13] Additionally, the effort to adapt them for MD or Monte Carlo simulations requires expertise and hardware knowledge typically beyond the skills or time of practitioners. The need to improve the scalability of mesh methods for electrostatics in MD simulation softwares have led to varied strategies such as performing the FFT on a subset of processors as in GROMACS[14] and the development of specific FFT software as in DL_POLY[15,16] and LAMMPS.[17,18] These all still suffer from the poor scaling properties of the FFT. One alternative to PME is the multilevel summation method (MSM).[19] While MSM is promising, currently, it has not been shown to scale better than particle mesh Ewald or particle−particle particle−mesh (P3M).[19,20] Another promising alternative is the isotropic periodic sum method,[21,22] but this is designed for homogeneous systems, and as a result, it is not very efficient for heterogeneous systems.

In this paper, we develop and present a mesh-free hierarchical clustering algorithm called the periodic Coulomb treecode (PCT) method as an alternative to particle mesh Ewald in parallel simulations. The performance of the treecode method, unlike mesh-based methods, is independent of the distribution of the particles. The method is simple and suitable for free-space, 3D, 2D, and 1D periodic boundary conditions. In addition, it lays the framework for the implementation of previous works on higher-order electrostatic multipoles[23,24] in simulation packages such as TINKER,[25] AMBER,[26] and DL_POLY.[27] As a proof of concept, we incorporate our method into the simulation package DL_POLY Classic[28] and compare structural and dynamical properties of (i) water, (ii) water with an Na^+ ion and Cl^- ion, and (iii) valinoymycin in water, obtained via MD simulations using PME and PCT. We also provide parallel timing comparisons of PME and PCT to shed light on the potential efficiency of PCT. DL_POLY Classic uses a replicated data strategy[29] for load-balancing, which makes it suitable for the tree algorithm.

The rest of the paper is organized as follows: Section 2 introduces and develops the periodic Coulomb tree method. Section 3 explains the modifications made to DL_POLY Classic in order to implement PCT in the package. Section 4 provides simulation and timing results for comparison of PME to PCT and conclude with a summary and future directions.

2. PERIODIC COULOMB TREE METHOD (PCT)

The periodic Coulomb Tree method computes the electrostatic energy of the system directly via eq 1 using a Cartesian treecode algorithm. We provide a short introduction to the treecode algorithm and explain how we extend the method to periodic boundary conditions.

2.1. Treecode Algorithm. There are several variants of the treecode algorithm including particle−cluster,[30−39] cluster−particle,[36,40] and cluster−cluster methods.[41−44] Here, our implementation focuses on the particle−cluster variant. The algorithm hierarchically restructures the interacting particles into an oct-tree. The root of the tree is the smallest bounding box that encloses the particles. The root is divided into eight children (clusters), which form the first level of the tree. Each of these eight children (clusters) are then divided into eight children (clusters). The process continues recursively on each branch and terminates when the number of particles in a cluster is less than or equal to a predetermined number, N_0. The last cluster at the end is called a leaf. An example clustering in 2D is shown in Figure 1. In the example, the clustering terminates for $N_0 = 3$.

2.2. The Free-Space Coulomb Treecode. The treecode for the $\frac{1}{r}$ kernel has been well studied and implemented in several applications.[24,30,32,34,36,45] The free-space Coulomb treecode computes eq 1 for just the fundamental cell with no interactions with the periodic images. In this case, the energy is given by

$$U(r_1, ..., r_N) = \frac{1}{2}\sum_{i\neq j}^{N}\frac{q_i q_j}{|\mathbf{r_i} - \mathbf{r_j}|} = \frac{1}{2}\sum_{i=1}^{N} q_i \sum_{j\neq i}^{N}\frac{q_j}{|\mathbf{r_i} - \mathbf{r_j}|} = \frac{1}{2}\sum_{i=1}^{N} U_i \tag{5}$$

where U_i is the energy due to the interaction of particle i with all the other particles in the fundamental cell. The efficiency of the treecode is largely due to how U_i is computed. Let particle i, with charge q_i, be at position $\mathbf{x}_i$ interacting with all the other particles, which are at position $\mathbf{y}_j$, $\{\mathbf{y}_j, q_j, j = 1{:}N\}$. Then,

$$U_i = U(\mathbf{x}_i) = q_i \sum_{j\neq i}^{N}\frac{q_j}{|\mathbf{x}_i - \mathbf{y}_j|} = q_i \sum_{j\neq i}^{N} q_j \phi(\mathbf{x}_i, \mathbf{y}_j) \tag{6}$$

The treecode uses the hierarchical tree to separate the particle−particle interactions in eq 6 into near-field and far-field interactions. The particle−particle interactions are replaced by particle−cluster interactions and eq 6 becomes

$$U(\mathbf{x}_i) = q_i \sum_{j\neq i}^{N} q_j \phi(\mathbf{x}_i, \mathbf{y}_j) = q_i \sum_{C} \sum_{\mathbf{y}_j \in C} q_j \phi(\mathbf{x}_i, \mathbf{y}_j) \tag{7}$$

where C represents clusters. Particle−cluster interactions that are near field are computed directly while those that are far field are approximated by a Taylor approximation about the center of the cluster. A schematic of a particle−cluster interaction is shown in Figure 2. Cluster C has radius r and center $\mathbf{y}_c$, and the

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

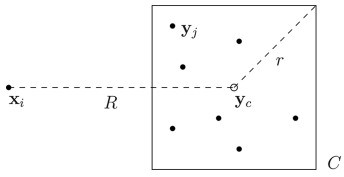


Figure 2. Interaction between a particle at $\mathbf{x}_i$ and particles at $\mathbf{y}_j$ in cluster C. The cluster has center $\mathbf{y}_c$ and radius r, and the particle−cluster distance is $R = |\mathbf{x}_i - \mathbf{y}_c|$.

distance from $\mathbf{x}_i$ to $\mathbf{y}_c$ is R. A particle−cluster interaction is considered to be far field or well separated when $\frac{r}{R} \leq \theta$, where $0 < \theta < 1$.

2.2.1. Approximations for Far-Field Interactions. For a well-separated particle−cluster interaction, the potential energy due to the interactions is approximated by a pth order three-dimensional Taylor polynomial given as

$$U(\mathbf{x}_i) = q_i \sum_{\mathbf{y}_j \in C} q_j \phi(\mathbf{x}_i, \mathbf{y}_j) \approx q_i \sum_{\mathbf{y}_j \in C} q_j \sum_{\|\mathbf{k}\|=0}^{p} \frac{1}{\mathbf{k}!} \partial_{\mathbf{y}}^{\mathbf{k}} \phi(\mathbf{x}_i, \mathbf{y}_c)(\mathbf{y}_j - \mathbf{y}_c)^{\mathbf{k}} \quad (8)$$

$$= q_i \sum_{\|\mathbf{k}\|=0}^{p} \frac{1}{\mathbf{k}!} \partial_{\mathbf{y}}^{\mathbf{k}} \phi(\mathbf{x}_i, \mathbf{y}_c) \sum_{\mathbf{y}_j \in C} q_j (\mathbf{y}_j - \mathbf{y}_c)^{\mathbf{k}} \quad (9)$$

$$= q_i \sum_{\|\mathbf{k}\|=0}^{p} a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) M_{\mathbf{k}}(C) \quad (10)$$

where $\mathbf{k} = \langle k_1, k_2, k_3 \rangle$, $\|\mathbf{k}\| = k_1 + k_2 + k_3$, $\mathbf{k}! = k_1!k_2!k_3!$, $\partial_{\mathbf{y}}^{\mathbf{k}} = \partial_{y_1}^{k_1} \partial_{y_2}^{k_2} \partial_{y_3}^{k_3}$, and $(\mathbf{y}_j - \mathbf{y}_c)^{\mathbf{k}} = (y_{j_1} - y_{c_1})^{k_1} (y_{j_2} - y_{c_2})^{k_2} (y_{j_3} - y_{c_3})^{k_3}$. Eq 10 is a pth order Taylor approximation for the particle−cluster interaction with Taylor coefficients

$$a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) = \frac{1}{\mathbf{k}!} \partial_{\mathbf{y}}^{\mathbf{k}} \phi(\mathbf{x}_i, \mathbf{y}_c) \quad (11)$$

and cluster moments

$$M_{\mathbf{k}}(C) = \sum_{\mathbf{y}_j \in C} q_j (\mathbf{y}_j - \mathbf{y}_c)^{\mathbf{k}} \quad (12)$$

We note that the moments defined above have no dependence on the target particle i. As such, the moments for each cluster can be computed, stored, and reused in eq 10 for all far-field interactions of the cluster in question with different target particles.

From eqs 8−10, the force on particle i due to the particles in a far-field cluster C is approximated as

$$\mathbf{f}_{i,C} = -q_i \nabla_{\mathbf{x}_i} \sum_{\mathbf{y}_j \in C} q_j \phi(\mathbf{x}_i, \mathbf{y}_j) \approx q_i \sum_{\|\mathbf{k}\|=0}^{p} \nabla_{\mathbf{y}} a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) M_{\mathbf{k}}(C) \quad (13)$$

$$= q_i \sum_{\|\mathbf{k}\|=0}^{p} \begin{bmatrix} \partial_{y_1} a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) \\ \partial_{y_2} a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) \\ \partial_{y_3} a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) \end{bmatrix} M_{\mathbf{k}}(C) \quad (14)$$

$$= q_i \sum_{\|\mathbf{k}\|=0}^{p} \begin{bmatrix} (k_1 + 1) a_{\mathbf{k}+\mathbf{e}_1} \\ (k_2 + 1) a_{\mathbf{k}+\mathbf{e}_2} \\ (k_3 + 1) a_{\mathbf{k}+\mathbf{e}_3} \end{bmatrix} M_{\mathbf{k}}(C) \quad (15)$$

where $\mathbf{e}_1 = \langle 1,0,0 \rangle$, $\mathbf{e}_2 = \langle 0,1,0 \rangle$, and $\mathbf{e}_3 = \langle 0,0,1 \rangle$. From eqs 14 to 15, we have applied eq 11 to get

$$\partial_{y_i} a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) = \frac{1}{\mathbf{k}!} \partial_{y_i} \partial_{\mathbf{y}}^{\mathbf{k}} \phi(\mathbf{x}_i, \mathbf{y}_c) = \frac{(k_i + 1)}{(k_i + 1)} \frac{1}{\mathbf{k}!} \partial_{\mathbf{y}}^{\mathbf{k}+\mathbf{e}_i} \phi(\mathbf{x}_i, \mathbf{y}_c)$$

$$= (k_i + 1) \frac{1}{(\mathbf{k} + \mathbf{e}_i)!} \partial_{\mathbf{y}}^{\mathbf{k}+\mathbf{e}_i} \phi(\mathbf{x}_i, \mathbf{y}_c)$$

$$= (k_i + 1) a_{\mathbf{k}+\mathbf{e}_i} \quad (16)$$

We apply the force approximation in eq 15 to approximate a far-field electrostatic contribution to the virial as

$$V_s = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} \mathbf{f}_{ij} \cdot \mathbf{r}_{ij} = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in C} \mathbf{f}_{i,C} \cdot \mathbf{r}_{ij} \approx \frac{1}{2} \sum_{C} \sum_{i=1}^{N} \mathbf{f}_{i,C} \cdot \mathbf{r}_{i,C} \quad (17)$$

where $\mathbf{r}_{i,C}$ is the distance from particle i to the center of cluster C.

To avoid explicit formulas for higher-order derivatives, the Taylor coefficient $a_{\mathbf{k}}$ is computed by the recurrence relation

$$a_{\mathbf{k}}(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{x} - \mathbf{y}|^2} \left\{ \left(2 - \frac{1}{\|\mathbf{k}\|}\right) \sum_{i=1}^{3} (x_i - y_i) a_{\mathbf{k}-\mathbf{e}_i} + \left(\frac{1}{\|\mathbf{k}\|} - 1\right) \sum_{i=1}^{3} a_{\mathbf{k}-2\mathbf{e}_i} \right\} \quad (18)$$

where $\mathbf{x} = \langle x_1, x_2, x_3 \rangle$ and $\mathbf{y} = \langle y_1, y_2, y_3 \rangle$. The starting value for the recurrence is the coefficient for $\|\mathbf{k}\| = 0$, which is computed explicitly. The procedure sets $a_{\mathbf{k}} = 0$ when any of the indices is negative. In our implementation, $\mathbf{x} = \mathbf{x}_i$ and $\mathbf{y} = \mathbf{y}_c$. The derivation for the recurrence above has been provided by previous authors.[32,34]

2.2.2. Pseudo-Code for the Parallel Free-Space Treecode. Algorithm 1 below is a pseudo-code for the parallel free-space particle−cluster treecode. The parallelization is via a replicated data strategy with MPI. The user specifies the order of Taylor approximation, p, the multipole acceptability criterion (MAC), θ, and the maximum number of particles in a leaf, N_0. Because each processor stores all the particles, each builds a copy of the tree. The particle−cluster interactions are then split over the P processors. A processor with label $l \in \{0, ..., P-1\}$computes all the particle−cluster interactions for particles numbered $A_1 = \frac{l \times N}{P} + 1$ to $A_2 = \frac{(l+1) \times N}{P}$. The algorithm loops over particles A_1 to A_2, and calls the subroutine evaluate-fs-pc in line 5 to perform a particle−root interaction for each particle. The MAC is not satisfied for any of the particle−root interaction; hence, the algorithm descends to the next level of the tree, that is, the children of the current cluster. On this level, the MAC is tested again for each potential interaction between the particle and the clusters on this level of the tree. If the MAC is satisfied for a particular particle−cluster pair, the algorithm performs two tasks:

1. If the moments of the current cluster is not available, it evaluates and stores the moments using eq 12,
2. It approximates the particle−cluster interaction using the Taylor series polynomial in eqs 10 and 15.

If the MAC fails for a particle−cluster pair, the algorithm descends to the next level and checks the children of the cluster. If the MAC keeps failing, the procedure will continue until the algorithm reaches a leaf. The particle−leaf interaction is then evaluated by direct summation.

A global sum of the potential energy is performed at the end of each computation.

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Algorithm 1. Free-space treecode

```
program fs-treecode
  input : N particles x_i, charges q_i, parameters p, θ, N_0
  output : energies, forces
  Set {y_j}_{j=1}^N = {x_i}_{j=1}^N; construct tree of source particles y_j
  for i = A_1 : A_2; evaluate-fs–pc(x_i,root)
end program
subroutine evaluate-fs-pc (x, C)
  if MAC is satisfied, i.e. (r/|x − y_c| ≤ θ)
    if not available, evaluate and store moments of C
    evaluate particle-cluster interaction by eqns (10), (15) and (17)
  else if C is a leaf
    use direct sum to evaluate particle-leaf interaction
  else for each child of C
    evaluate-fs-pc(x,child)
end subroutine
```

2.3. The Periodic Coulomb Tree. The focus of this paper is to present the periodic Coulomb tree (PCT) method as a viable alternative to particle mesh Ewald for parallel simulations using replicated data strategy. PCT extends the free-space treecode to periodic boundary conditions. It approximates eq 1 over a given finite set of periodic images. Similar to the development of the free-space treecode, we consider a particle i, with charge q_i, at position $\mathbf{x}_i$ interacting with particles $\mathbf{y}_j$, $\{\mathbf{y}_j, q_j = 1{:}N\}$, and their periodic images. The energy due to this interaction can be written as

$$\begin{aligned} U_p(\mathbf{x}_i) &= q_i \sum_{\mathbf{n}}{}' \sum_{j=1}^{N} \frac{q_j}{|\mathbf{x}_i - \mathbf{y}_j + \mathbf{n}L|} \\ &= q_i \sum_{j\neq i}^{N} \frac{q_j}{|\mathbf{x}_i - \mathbf{y}_j|} + q_i \sum_{\mathbf{n}\neq\mathbf{0}} \frac{q_j}{|\mathbf{x}_i - \mathbf{y}_j + \mathbf{n}L|} \\ &= q_i \sum_{j\neq i}^{N} q_j \phi(\mathbf{x}_i, \mathbf{y}_j) + q_i \sum_{\mathbf{n}\neq\mathbf{0}} \sum_{j=1}^{N} q_j \phi(\mathbf{x}_i, \mathbf{y}_j + \mathbf{n}L) \end{aligned} \tag{19}$$

where the first summation is the energy due to interactions with particles in the fundamental cell, and the second summation is the energy due to interactions with the periodic images. Different periodic boundary conditions can be handled in PCT by simply specifying the correct range of $\mathbf{n} = (n_1, n_2, n_3)$ where $n_i, i \in \{1,2,3\}$, is an integer. In this work, we apply periodic boundary conditions in all three directions, as such, all three coordinates of $\mathbf{n}$ vary. To specify periodic boundary conditions in two dimensions, one coordinate is held constant, and the other two vary. Periodic boundary conditions in one dimension is achieved by holding two coordinates constant and varying the third. The algorithm defaults to the free-space boundary condition when $\mathbf{n} = 0$, which means the second summation in eq 19 vanishes.

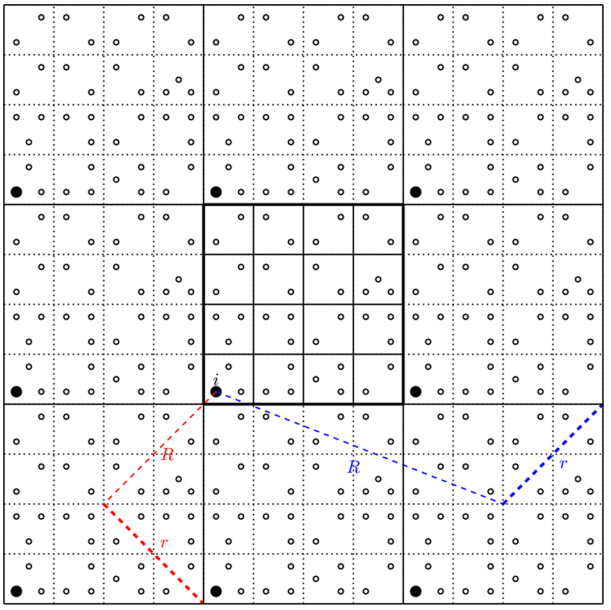


Figure 3. A schematic of a 2D periodic tree showing a hierarchically clustered fundamental cell and its nearest neighbors. The maximum number of particles in a leaf is $N_0 = 3$.

2.3.1. Approximations for Far-Field Interactions. In the periodic tree algorithm, the fundamental cell is hierarchically clustered into the tree, and this clustering is maintained in the periodic replicas. Figure 3 is a schematic of a 2D periodic tree showing a fundamental cell hierarchically clustered up to level 2 and the nearest neighbor periodic images. To approximate $U_p(\mathbf{x}_i)$ in eq 19, observe that the first summation is exactly the free-space energy, $U(\mathbf{x}_i)$, which is approximated by the pth order Taylor approximation in eq 10. The second sum is the energy due to the interaction of particle i in the fundamental cell with the particles in the periodic images. This sum is also split into a near-field and a far-field interaction. Thus, the far-field interaction is approximated as

$$q_i \sum_{\mathbf{n}=\mathbf{0}} \sum_{j=1}^{N} q_j \phi(\mathbf{x}_i, \mathbf{y}_j + \mathbf{n}L) = q_i \sum_{\mathbf{n}=\mathbf{0}} \sum_{\mathbf{y}_j \in C} q_j \phi(\mathbf{x}_i, \mathbf{y}_j + \mathbf{n}L) \tag{20}$$

$$\approx q_i \sum_{\mathbf{n}=\mathbf{0}} \sum_{\mathbf{y}_j \in C} \sum_{\|\mathbf{k}\|=0}^{p} q_j \frac{1}{\mathbf{k}!} \partial_{\mathbf{y}}^{\mathbf{k}} \phi(\mathbf{x}_i, \mathbf{y}_c + \mathbf{n}L)[\mathbf{y}_j + \mathbf{n}L - (\mathbf{y}_c + \mathbf{n}L)]^{\mathbf{k}} \tag{21}$$

$$= q_i \sum_{\mathbf{n}=\mathbf{0}} \sum_{\|\mathbf{k}\|=0}^{p} \frac{1}{\mathbf{k}!} \partial_{\mathbf{y}}^{\mathbf{k}} \phi(\mathbf{x}_i, \mathbf{y}_c + \mathbf{n}L) \sum_{\mathbf{y}_j \in C} q_j (\mathbf{y}_j - \mathbf{y}_c)^{\mathbf{k}} \tag{22}$$

$$\approx q_i \sum_{n_3=-s}^{s} \sum_{n_2=-s}^{s} \sum_{n_1=-s}^{s} \sum_{\|\mathbf{k}\|=0}^{p} a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c + \mathbf{n}L) M_{\mathbf{k}}(C) \tag{23}$$

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

where in eq 23 the sum over infinite periodic images is replaced by a sum over finite periodic images determined by user input $s \geq 0$. For example, $s = 0$ limits the algorithm to only interactions in the fundamental cell while $s = 1$ includes interactions with the nearest periodic images. The key time and memory-saving idea underlying the periodic tree is that the moments for the periodic clusters in eq 23 are the same as the moments for the clusters in the fundamental cell. Hence, there is no need to build and store the periodic images. Because the distance between particles in an image is the same as in the fundamental cell, the hierarchical tree for each image is the same as the hierarchical tree for the fundamental cell. As such, the algorithm only builds the tree for the fundamental cell. The interactions with the periodic images reduce to a shifted interaction with the fundamental cell. The only new evaluation required is to compute the multidimensional Taylor coefficients $a_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c + \mathbf{n}L)$. The far-field force on particle i is approximated as

$$\mathbf{f}_{i,p,C} \approx q_i \sum_{n_3=-s}^{s} \sum_{n_2=-s}^{s} \sum_{n_1=-s}^{s} \sum_{\|\mathbf{k}\|=0}^{p} \begin{bmatrix} (k_1+1)a_{\mathbf{k}+\mathbf{e}_1}(\mathbf{x}_i, \mathbf{y}_c + \mathbf{n}L) \\ (k_2+1)a_{\mathbf{k}+\mathbf{e}_2}(\mathbf{x}_i, \mathbf{y}_c + \mathbf{n}L) \\ (k_3+1)a_{\mathbf{k}+\mathbf{e}_3}(\mathbf{x}_i, \mathbf{y}_c + \mathbf{n}L) \end{bmatrix} M_{\mathbf{k}}(C) \tag{24}$$

Another computationally efficient feature of the periodic tree is that the interaction between particles in the fundamental cell and the periodic images happens at higher levels of the tree for periodic images that are farther away. As an example, consider the large dark filled particle, labeled i, in Figure 3. To fix ideas, we focus on the interaction of particle i with the periodic images in the lower left and lower right nearest neighbors (clusters). The initial interaction of i with the lower left cluster fails the multipole acceptability criterion since $\theta < \frac{r}{R} \approx 1$. The algorithm then descends to the lower level, and the method proceeds recursively. On the other hand, the initial interaction with the lower right cluster, which is farther from particle i, will pass the multipole acceptability criterion since $\frac{r}{R} < 0.5 < \theta$. Thus, particle i interacts once with the lower right periodic image with no recursive calls to lower level clusters.

2.3.2. Pseudo-Code for the Parallel Periodic Coulomb Tree Method. The PCT algorithm is similar to the free-space algorithm. It takes in the same inputs as the free-space algorithm as well as the additional parameter, s, which determines the number of periodic images to be included in the computation. The particles are split over the processors in the same way as the free-space method. However, for PCT, in addition to the loop over particles, there is an additional loop over the periodic images. As such, line 5 of Algorithm 1 is expanded to include a loop over the periodic images. The pseudo-code for PCT is given in Algorithm 2.

We implemented the periodic PCT method in DL_POLY Classic. The details of the implementation are provided in Section 1 of the Supporting Information.

Algorithm 2. Periodic Coulomb Tree Method

```
1  program PCT
2    input : N particles x_i, charges q_i, parameters p, θ, N_0, s
3    output : energies, forces
4    Set {y_j}_{j=1}^N = {x_i}_{j=1}^N; construct tree of source particles y_j
5    for i = A_1 : A_2
6      for n_3 = −s : s; n_2 = −s : s; n_1 = −s : s
7        n = (n_1, n_2, n_3); evaluate-ppc(x_i,n, root)
8  end program
9  subroutine evaluate-ppc (x, n, C)
10   if MAC is satisfied, i.e. ( r / |x − (y_c + nL)| ≤ θ )
11     if the moments of C are not available, evaluate and store the moments
12     evaluate the particle-cluster interaction by equations (23) and (24)
13   else if C is a leaf
14     use direct sum to evaluate particle-leaf interaction
15   else for each child of C
16     evaluate-ppc(x,n, child)
17 end subroutine
```

3. NUMERICAL RESULTS

Our goal in this section is to provide numerical justification for the efficiency of PCT. We compare PME and PCT results from NVT ensemble MD simulations for three systems:

1. A total of 6912 molecules of SPC water in a cubic cell of length 69.5 Å at 307.46 K with a Nosé−Hoover thermostat;
2. Sodium chloride in 139 molecules of SPC water in a cubic cell of length 17.9 Å at 295 K with an Evans Gaussian thermostat;
3. Valinomycin molecule in 1223 molecules of SPC water in a truncated octahedral cell with width 49.9 Å at 310 K, also with an Evans Gaussian thermostat.

Further details on the systems are given in the Supporting Information.

3.1. System 1. We compare PME and PCT results for structural and dynamical properties of an NVT ensemble MD simulation of 6912 molecules of SPC water in a $(69.5\ \text{Å})^3$ box at 307.46 K.[46,47]

In addition to the results on structural and dynamical properties, we provide timing comparisons for a system of N = 20,736 atoms (6912 water molecules) and N = 49,152 atoms (16,384 water molecules) on up to 512 and 1024 parallel cores, respectively.

3.1.1. Radial Distribution Functions. In Figure 4, we compare the O−O, O−H, and H−H radial distribution functions for both PME and PCT with order of approximation

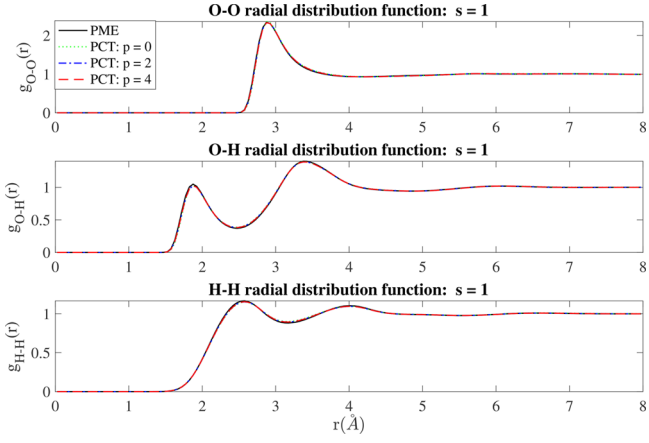


Figure 4. The O−O (top), O−H (middle), and H−H (bottom) radial distribution functions for PME and PCT with $p \in \{0,2,4\}$ and $s = 1$.

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

$p \in \{0,2,4\}$. The lattice sum for PCT in Figure 4 includes the fundamental cell and the nearest neighbor images, that is, $s = 1$ and $p \in \{0,2,4\}$ in eqs 23 and 24.

From the plots, we see that the radial distribution functions obtained using PCT match well with those for PME for all orders of PCT.

3.1.2. Velocity and Force Autocorrelation Functions. Figure 5 is a comparison of the unnormalized center of mass velocity

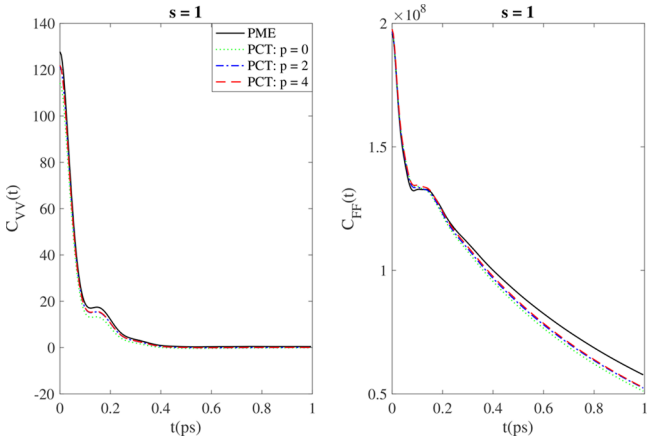


Figure 5. Center of mass velocity (left) and force (right) autocorrelation functions for PME and PCT with $p \in \{0,2,4\}$ and $s = 1$.

and force autocorrelation functions obtained via PME and PCT. The unnormalized functions highlight the differences in the magnitude of the initial correlation for PME and PCT. The forces computed by the two methods necessarily differ because the potential energy functions are different. PME approximates eqs 2 and 3 using a cutoff in real space and a finite set of the infinite reciprocal space **k** vectors, respectively, while PCT approximates eq 1 using a finite set of the infinite real space **n** vectors and far-field Taylor approximations. We see from Figure 5 that PME and PCT produce similar dynamics. The peaks and valleys in the functions occur at the same times. The rate of decay of the autocorrelation functions are the same for both PME and PCT. We do not expect perfect agreement between the functions because the potentials and the approximations are different.

A spectral analysis of the normalized correlation functions produce almost indistinguishable results between PME and PCT. The plots from the spectral analysis are given in Section 2.1 of the Supporting Information.

3.1.3. Mean Square Displacement. The mean square displacements of the oxygen and hydrogen atoms are shown in Figure 6. We computed the self-diffusion coefficient D_{H_2O} as an average of the diffusion coefficient of oxygen, D_O and hydrogen, D_H using the Einstein relation.[1] The results are given in Table 1.

These affirm the similarities in the dynamics produced by PME and PCT.

3.1.4. Distance-Dependent Kirkwood G-Factor $G_k(r)$. To give a more complete picture of PCT, we provide the results of computing the Kirkwood G-factor, $G_k(r)$, using PME and PCT with $s = 1$ in Figure 7 where

$$G_k(r) = \mu_i \cdot \sum_{r_{ij} \le r} \mu_j \tag{25}$$

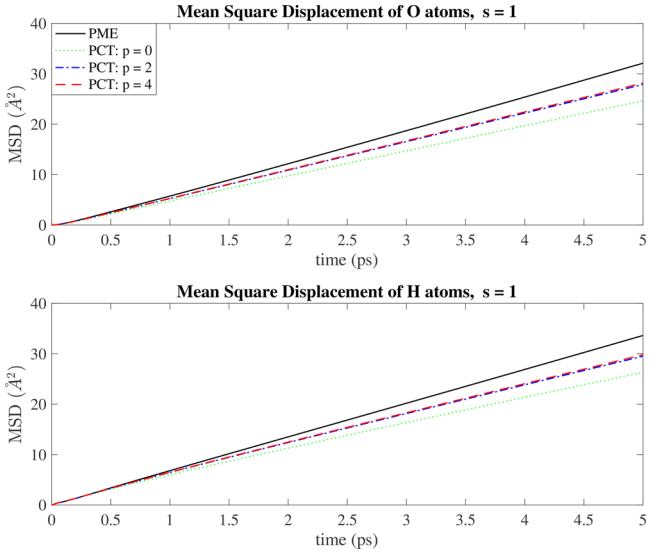


Figure 6. The mean square displacement for Oxygen (top) and hydrogen (bottom) atoms.

Table 1. Self-Diffusion Coefficients, $D_{H_2O} \pm 0.1$ (10^{-9} m^2 s^{-1}) of H_2O for PME and PCT

Method	D_O	D_H	D_{H_2O}
PME	11.1	11.2	11.2
PCT (p = 0)	8.3	8.3	8.3
PCT (p = 2)	9.4	9.5	9.5
PCT (p = 4)	9.5	9.6	9.5

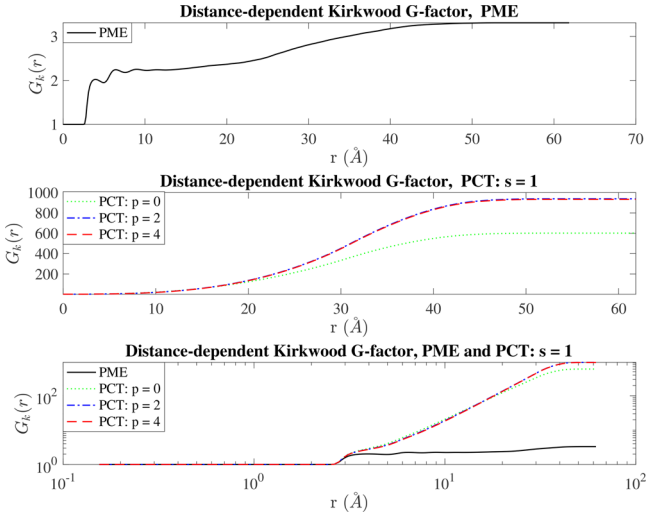


Figure 7. Distance-dependent Kirkwood G-factor for PME and PCT for system 1

and μ_i and μ_j are the unit vectors in the direction of molecule i and molecule j, respectively. As noted by several authors,[48−51] the Kirkwood G-factor depends on the method for computing the long-range electrostatic interactions. It is not surprising that $G_k(r)$ is different for PME and PCT in Figure 7. Despite this difference, both methods produce the same Kirkwood correlation length r_k = 53.656Å. The size of the electrostatic interaction cutoff has an effect on $G_k(r)$.[49,50] There is no cutoff for PCT, and setting $s \ge 1$ approximates an infinite system. We

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

will show the effect of periodic images on $G_k(r)$ in our results for system 2 in the next section.

3.1.5. Effect of Periodic Images. We also investigated the effect and importance of the number of periodic images included in the lattice sum for PCT. We performed MD simulations and computed the radial distribution and autocorrelation functions for $s \in \{0,1,2,3\}$. When $s = 0$, that means there are no periodic images, and we expect the surface effects to be significant. For $s = 1$, $s = 2$, and $s = 3$, there are 26, 124, and 342 periodic images, respectively. Figure 8 shows the radial distribution functions for

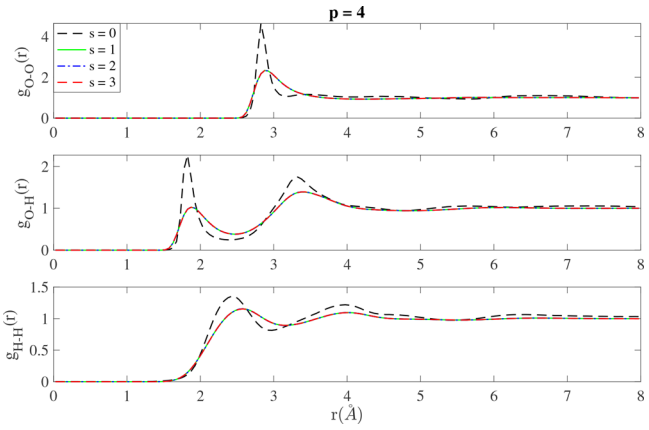


Figure 8. Radial distribution functions for different PCT lattice sums.

$p = 4$ and $s \in \{0,1,2,3\}$. We see that the $s = 0$ results deviate significantly from the $s > 0$ results. However, there is no significant difference between $s = 1$ and $s > 1$ for the system we studied. Thus, the most efficient computational choice is $s = 1$. Figure 9a,b is the plot of the unnormalized and normalized autocorrelation functions, respectively, for $p = 4$ and $s \in \{0,1,2,3\}$. The figures reinforce the importance and sufficiency of including the nearest neighbor periodic images. There is a big difference between $s = 0$ and $s > 0$ but little difference between $s = 1$ and $s \in \{2,3\}$.

3.2. System 2. Here, we compare PCT to PME for an NVT system of Na^+ and Cl^- ions in a solution of 139 SPC water molecules.

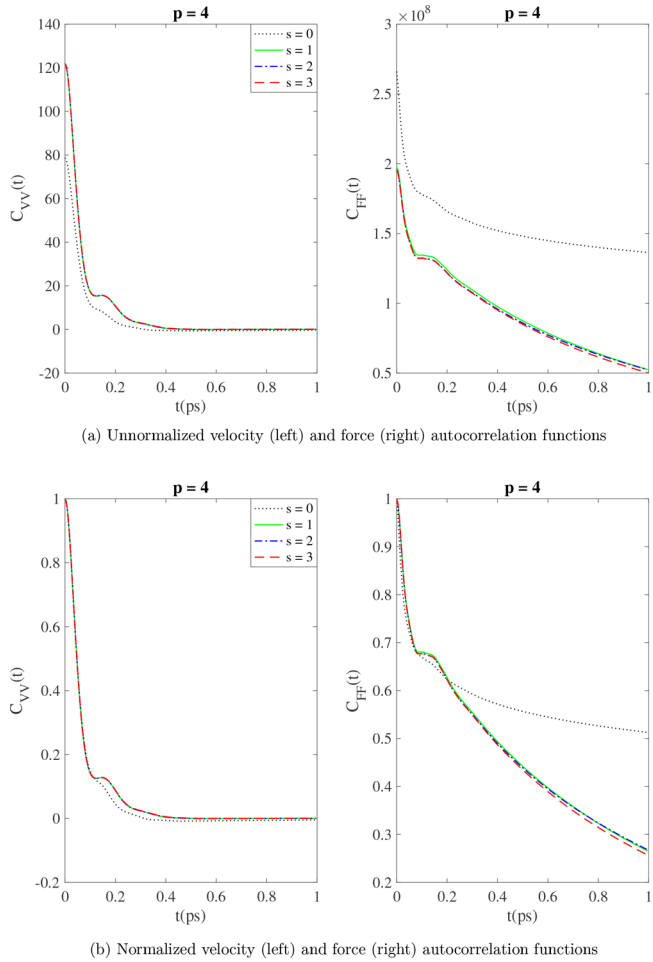
3.2.1. Radial Distribution Functions. The radial distribution functions from PME and PCT are compared in Figure 10. We see that PCT largely matches PME.

3.2.2. Distance-Dependent Kirkwood G-Factor $G_k(r)$. The Kirkwood G-Factor computed using only the water molecules is shown in Figure 11. The behavior is similar to the behavior observed for system 1 in Figure 7. For both PME and PCT, for all orders, the Kirkwood correlation length is $r_k = 15.5158$Å.

We also looked at the effect of periodic images on this system. The results, provided in Section 2.2 of the Supporting Information, reinforce our conclusion from system 1 that $s = 1$ is the most efficient choice for the periodic images in PCT.

3.3. System 3. The third system for comparison is a molecule of valinomycin in 1223 molecules of SPC water. The parameters of the system are described in a previous work by several authors.[52−55] Figure 12 shows the radial distribution function for several atom pairs for PME and PCT. The notation used in the plots of the radial distribution function is explained in Table 2.

We conclude from Figure 12 that PME and PCT produce qualitatively similar radial distribution functions for system 3.



(a) Unnormalized velocity (left) and force (right) autocorrelation functions

(b) Normalized velocity (left) and force (right) autocorrelation functions

Figure 9. Autocorrelation functions for different PCT lattice sums with $p = 4$ and $s \in \{0,1,2,3\}$. Left is velocity and right is force

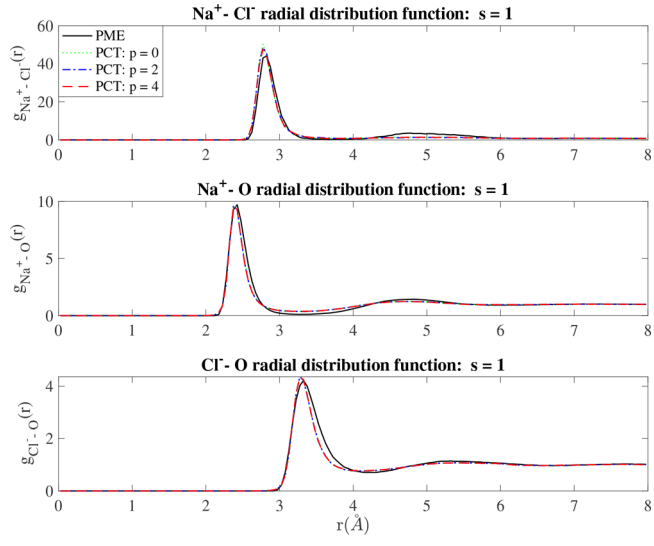


Figure 10. The Na^+-Cl^- (top), Na^+-O (middle) and Cl^--O (bottom) radial distribution functions

3.4. Timing Comparisons. The previous sections provide evidence that PCT produces results similar to PME. This section provides support for adopting PCT as an alternative to PME for parallel simulations employing replicated data strategy. We provide parallel timing comparisons for PME and PCT for a

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

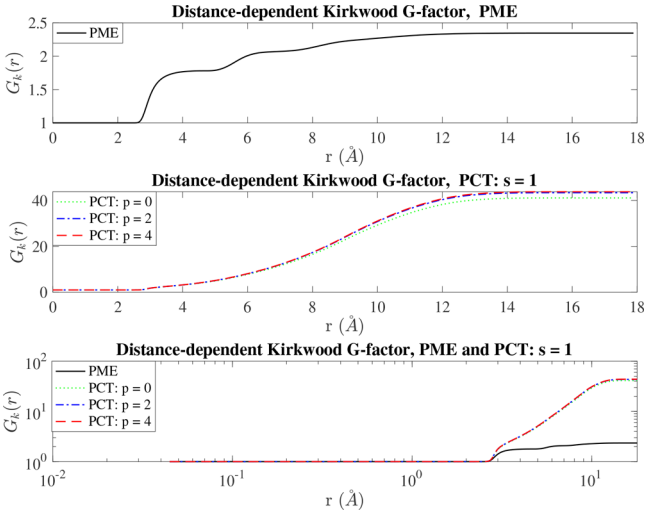


Figure 11. Distance-dependent Kirkwood G-factor for PME and PCT for system 2 computed using water molecules.

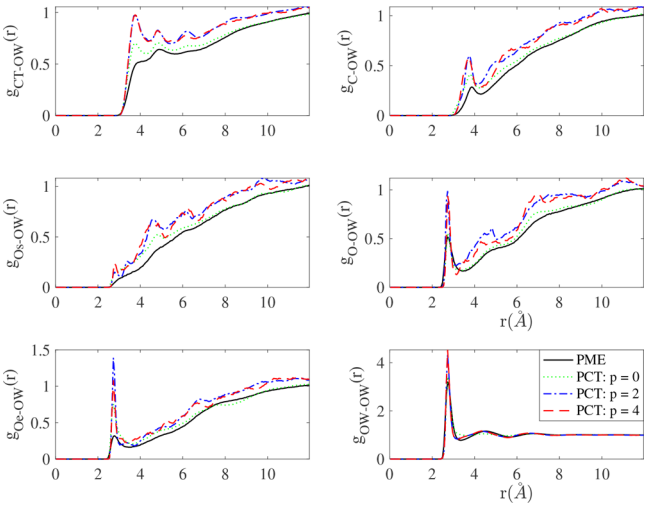


Figure 12. Radial distribution functions from the simulation of valinomycin in water for PME and PCT with $s = 1$.

Table 2. Notation for Radial Distribution Plots for System 3

Notation	Atom
OW	oxygen of a water molecule
CT	sp^3 hybridized carbon of valinomycin
C	sp^2 hybridized carbon of valinomycin
O	oxygen of the amide carbonyl
Oe	oxygen of the ester carbonyl
Os	the ester linkage oxygen

system of (a) N = 20,736 atoms, (b) N = 49,152 and (c) N = 393,216 atoms. The PME parameters for all three systems are the same with r_{cut} = 8.0 Å, α = 0.036037Å^{-1} and b-splines of order 8. The N = 20,736 atoms and N = 49,152 systems had a grid size of 32 × 32 × 32 while the N = 393,216 atoms had a grid size of 64 × 64 × 64. The timing runs were performed on Vesta, Argonne Leadership Computing Facility's test and development platform. Vesta is an IBM BG/Q with 2048 nodes and 16 cores per node. Each core has 1 GB RAM of memory. The interconnect is a 5D Torus Proprietary Network and the default FFT routine on Vesta is FFTW3 version 3.2.3.

Figure 13 is a plot of the CPU time per time step for PME and PCT for different processor counts obtained as an average over

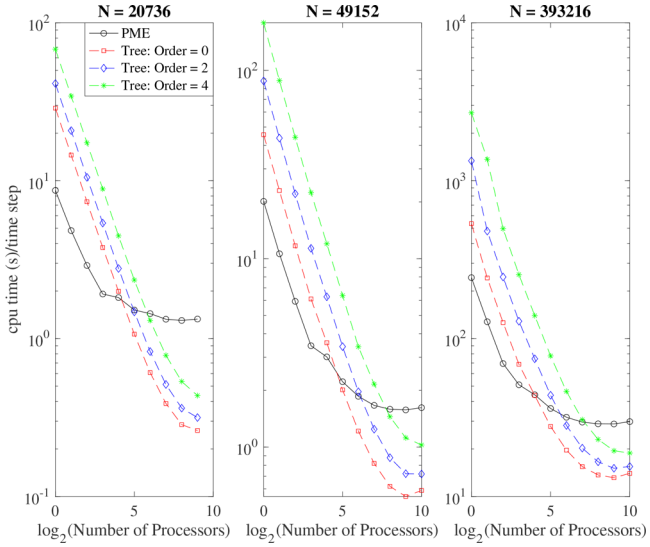


Figure 13. CPU time per time step for different processor counts for PME and PCT with s = 1.

100 MD steps for systems (a) and (b) and over 10 MD steps for system (c). The PCT plots are for s = 1 and order of approximation $p \in \{0,2,4\}$. We see that although PME is very efficient for a small number of processors, the communication overhead results in poor parallel scaling on 16 processors or more. The communication overhead for PCT is less severe, and it becomes more efficient than PME for a higher processor size. Table 3 shows the data for the plots in Figure 13. We see that for N = 20,736, PCT provides 5, 4, and 3 times the speedup over PME for multipole orders p = 0, p = 2, and p = 4, respectively. For N = 49,152, the speedup provided is 3, 2, and 1.5 times for p = 0, p = 2, and p = 4, respectively. For N = 393,216, we see approximately 2 times the speedup for p = 0 and p = 2 and 1.5 times the speedup for p = 4 in spite of the fact that the efficiency of the replicated data parallelization strategy implemented in DL_POLY Classic decays with an increasing system size. Thus, for applications where low order approximation suffices, PCT can provide a substantial speedup over PME.

In Section 2.4 of the Supporting Information, we provide timing comparisons for s = 2 and s = 3 for N = 20,736 and N = 49,152. As expected, PCT is less efficient in these regimes but it still offers some speedup over PME for high processor counts.

4. CONCLUSIONS

We developed the periodic Coulomb tree (PCT) method as an alternative to particle mesh Ewald (PME) for parallel MD simulations. PCT unlike mesh methods is suitable for both uniform and nonuniform particle distributions. In addition, it is easily adapted for simulations with periodic boundary conditions in one, two, or three dimensions as well as simulations with free-space boundary conditions.

Results from our MD simulations of liquid water showed that PCT produces similar results to PME. Having verified the accuracy of PCT, we provided parallel timing comparisons of PCT to PME. We showed that for moderate sized systems, PCT can provide up to five times the speedup over PME for parallel simulations using replicated data strategy.

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Table 3. CPU Time (s) per Time Step for Processor Sizes $\{2^i, i = 0{:}1{:}10\}$ for PME and PCT for System Size at N = 20,736 Particles, N = 49,152 Particles, and N = 393,216 Particles

processor size	PME	PCT: $p = 0$	PCT: $p = 2$	PCT: $p = 4$
		N = 20,736, s = 1		
1	8.67805	28.8081	41.2520	68.0480
2	4.83136	14.5215	20.7628	34.3301
4	2.90468	7.35779	10.4997	17.3771
8	1.91498	3.77268	5.38810	8.86696
16	1.81884	1.99112	2.77959	4.46939
32	1.51644	1.06732	1.47566	2.34844
64	1.43934	0.60921	0.82806	1.30177
128	1.32605	0.38817	0.51235	0.78323
256	1.30275	0.28508	0.36248	0.53504
512	1.32969	0.26176	0.31615	0.43552
		N = 49,152, s = 1		
1	20.1529	45.5423	87.8054	178.743
2	10.6257	23.0041	43.6811	88.2220
4	5.93655	11.7137	22.1210	44.0415
8	3.45139	6.11345	11.3493	22.4288
16	3.01048	3.58205	6.27550	12.0046
32	2.22345	2.01283	3.41449	6.37295
64	1.86014	1.21649	1.96697	3.40867
128	1.66483	0.81914	1.24471	2.15903
256	1.58822	0.61903	0.87862	1.44950
512	1.57578	0.54681	0.72271	1.12031
1024	1.61940	0.58861	0.71986	1.02650
		N = 393,216, s = 1		
1	243.250	534.736	1335.93	2693.18
2	127.851	242.189	479.366	1365.92
4	69.4414	126.249	245.519	496.416
8	51.0250	68.8101	128.836	254.327
16	44.0477	43.6415	74.4847	139.715
32	36.1062	27.7717	43.7535	77.6482
64	31.7569	19.6681	28.1768	46.3209
128	29.5851	15.4576	20.2416	30.6539
256	28.8597	13.6892	16.5793	22.9834
512	28.7980	13.1535	15.0942	19.4760
1024	29.7983	14.0134	15.4812	18.8424

The current work was developed solely for CPUs. We recognize that, increasingly, MD algorithms[56−59] are being developed for GPUs to take advantage of the speedup they offer over CPUs. Several authors have worked on efficient GPU implementations of treecode algorithms[60−63] with speedup of up to 60 times over comparable CPU implementations. A future direction is to develop a GPU version of PCT.

The work presented here was motivated by the need to circumvent the communication bottleneck of the FFT that is necessary for PME. This bottleneck is more pronounced for advanced potential energy surfaces involving multipolar electrostatics where the reciprocal space grids are larger. We believe extensions of the work presented here has the potential to address the increased communication overhead for multipolar electrostatic interactions.

A.1. Systems with a Net Charge

Our presentation assumed that our system was electrically neutral. Here, we outline an approach for handling systems with net charge in a box of length L. Let

$$q_{\text{tot}} = \sum_{i=1}^{N} q_i \tag{26}$$

$f(\mathbf{x}) = \frac{\alpha^3}{\pi^{3/2}} e^{-\alpha^2|\mathbf{x}|^2}$ and define a non-uniform unnormalized screening charge density

$$\rho_{\text{scr}}(\mathbf{x}) = -q_{\text{tot}} \cdot |\mathbf{x}| \cdot f(\mathbf{x}) \tag{27}$$

The Coulomb energy due to the interaction between a charge q_i at position $\mathbf{x}$ in the fundamental cell, $\mathbf{n} = 0$, and the screening charge density is

$$U_{i,\rho_{\text{scr}}} = \int_{\text{cell}} \frac{q_i}{|\mathbf{x}|} \rho_{\text{scr}}(\mathbf{x}) d\mathbf{x} = -q_{\text{tot}} \int_{\text{cell}} q_i f(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{28}$$

and the total potential energy in the fundamental cell and all periodic images is

$$U_{\rho_{\text{scr}}} = -\frac{1}{2} \sum_{\mathbf{n}} \sum_{i=1}^{N} \int_{\text{cell}} \frac{q_i}{|\mathbf{x} + \mathbf{n}L|} \rho_{\text{scr}}(\mathbf{x} + \mathbf{n}L) \mathrm{d}\mathbf{x} \tag{29}$$

$$= -\frac{q_{\text{tot}}}{2} \cdot \sum_{i=1}^{N} q_i \cdot \int_{\text{cell}} \sum_{\mathbf{n}} f(\mathbf{x} + \mathbf{n}L) \mathrm{d}\mathbf{x} \tag{30}$$

$$= -\frac{q_{\text{tot}}}{2} \cdot \sum_{i=1}^{N} q_i \cdot \int_{\text{cell}} \sum_{\mathbf{n}} f(\mathbf{x} + \mathbf{n}L) \mathrm{d}\mathbf{x} \tag{31}$$

$$= -\frac{q_{\text{tot}}^2}{2} \cdot \frac{\alpha^3}{\pi^{3/2}} \int_{\mathbf{R}^3} e^{-\alpha^2|\mathbf{x}|^2} \mathrm{d}\mathbf{x} \tag{32}$$

$$= -\frac{q_{\text{tot}}^2}{2} \tag{33}$$

From eqs 30 to 31, we have used the fact that integrating over the fundamental cell and all the periodic images is equivalent to an integration over all space. Also from eqs 32 to 33, we used the identity $\int_{\mathbf{R}^3} e^{-\alpha^2|\mathbf{x}|^2} \mathrm{d}\mathbf{x} = \frac{\pi^{3/2}}{\alpha^3}$. Thus, the Coulombic energy for a system with a net charge is given by eq 1 plus $U_{\rho_{\text{scr}}}$ given by eq 33. For a system with zero charge, no correction is needed since $q_{\text{tot}} = 0$ and $U_{\rho_{\text{scr}}} = 0$.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.9b00648. The information includes:

Details on the implementation of periodic PCT in DL_POLY Classic; details on parameters used in simulating the three systems; comparison of the spectral analysis of the normalized center-of-mass velocity and force autocorrelation functions for system 1 for both PCT and PME, providing further evidence of the similarity of PCT to PME; a study of the effect of periodic images on the radial distribution function and distance-dependent Kirkwood-G factor for system 2 where the results provide further evidence that using only nearest-neighbor periodic images (s = 1) in simulations is sufficient; and parallel timing comparisons of PME and PCT with s = 2 and s = 3 where the results show that PCT is competitive with PME even when more images are used in the simulation (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: boateng@sfsu.edu.

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

ORCID

Henry A. Boateng: 0000-0002-1066-149X

Notes

The author declares no competing financial interest.

■ ACKNOWLEDGMENTS

This work was partially supported by NSF grant CHE 1800181. We used computing resources provided through the Director's Discretionary Program of the Argonne Leadership Facility, which is a DOE Office of Science User Facility supported under contract DE-AC02-06CH11357. We would also like to thank an anonymous referee for very constructive feedback.

■ REFERENCES

(1) Allen, M. P.; Tildesley, D. J. *Computer simulation of liquids*; 1st Ed.; Oxford University Press, 1987.

(2) Ewald, P. P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.* **1921**, *369*, 253−287.

(3) Smith, W. *Elements of Molecular Dynamics*; 2014; Vol. *1*, http://scc.acad.bg/ncsa/articles/library/Library2016_Supercomputers-at-Work/Molecular_Dynamics/Elements-of-Molecular-Dynamics-Smith%20.pdf.

(4) Perram, J. W.; Petersen, H. G.; De Leeuw, S. W. An algorithm for the simulation of condensed matter which grows as the 3/2 power of the number of particles. *Mol. Phys.* **1988**, *65*, 875.

(5) York, D.; Yang, W. The fast Fourier Poisson method for calculating Ewald sums. *J. Chem. Phys.* **1994**, *101*, 3298.

(6) Darden, T.; York, D.; Pedersen, L. Particle mesh Ewald: An Nlog(N) method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98*, 10089.

(7) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* **1995**, *103*, 8577−8593.

(8) Kholmurodov, K.; Smith, W.; Yasuoka, K.; Darden, T.; Ebisuzaki, T. A SmoothParticle Mesh Ewald Method for DL_POLY Molecular Dynamics Simulation Package on the Fujitsu VPP700. *J. Comput. Chem.* **2000**, *21*, 1187−1191.

(9) Toukmaji, A.; Sagui, C.; Board, J.; Darden, T. Efficient particle-mesh Ewald based approach to fixed and induced dipolar interactions. *J. Chem. Phys.* **2000**, *113*, 10913−10927.

(10) Takahashi, D. *Lecture Notes in Comput. Sc.*; Springer: Berlin, 2009; Vol. *1*; pp 606−614.

(11) Ayala, O.; Wang, L.-P. Parallel implementation and scalability analysis of 3D Fast Fourier Transform using 2D decomposition. *Parallel Comput.* **2013**, *39*, 58−77.

(12) Pekurovsky, D. P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions. *SIAM J. Sci. Comput.* **2012**, *34*, C192−C209.

(13) Chan, A.; Balaji, P.; Gropp, W.; Rhakur, R. Communication Analysis of Parallel 3D FFT for Flat Cartesian Meshes on Large Blue Gene Systems. *Proceedings of the 15th International Conference on High Performance Computing*, Bangalore, India. 2008; pp 350−364.

(14) Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.* **2008**, *4*, 435−447.

(15) Bush, I. J.; Todorov, I. T.; Smith, W. A DAFT DL_POLY distributed memory adaptation of the Smoothed Particle Mesh Ewald method. *Comp. Phys. Commun* **2006**, *175*, 323−329.

(16) Todorov, I. T.; Smith, W.; Trachenko, K.; Dove, M. T. DL_POLY 3: New dimensions in molecular dynamics simulations via massive parallelism. *J. Mater. Chem.* **2006**, *16*, 1911−1918.

(17) Plimpton, S. J.; Pollock, R.; Stevens, M. Particle-Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations; *Eighth SIAM Conference on Parallel Processing for Scientific Computing* 1997.

(18) https://fftmpi.sandia.gov/papers.html.

(19) Hardy, D. J.; Wu, Z.; Phillips, J. C.; Stone, J. E.; Skeel, R. D.; Schulten, K. Multilevel summation method for electrostatic force evaluation. *J. Chem. Theory Comput.* **2015**, *11*, 766−779.

(20) Moore, S. G.; Crozier, P. S. Extension and evaluation of the multilevel summation methods for fast long-range electrostatics calculations. *J. Chem. Phys.* **2014**, *140*, 234112.

(21) Wu, X.; Brooks, B. R. Using the isotropic periodic sum method to calculate long-range interactions of heterogeneous systems. *J. Chem. Phys.* **2008**, *129*, 154115.

(22) Wu, X.; Brooks, B. R. The homogeneity condition: A simple way to derive isotropic periodic sum potentials for efficient calculation of long-range interactions in molecular simulation. *J. Chem. Phys.* **2019**, *150*, 214109.

(23) Albaugh, A.; Boateng, H. A.; Bradshaw, R. T.; Demerdash, O. N.; Dziedzic, J.; Mao, Y.; Margul, D. T.; Swails, J.; Zeng, Q.; Case, D. A.; Eastman, P.; Wang, L.-P.; Essex, J. W.; Head-Gordon, M.; Pande, V. S.; Ponder, J. W.; Shao, Y.; Skylaris, C.-K.; Todorov, I. T.; Tuckerman, M. E.; Head-Gordon, T. Advanced Potential Energy Surfaces for Molecular Simulation. *J. Phys. Chem. B* **2016**, *120*, 9811−9832.

(24) Boateng, H. A. Mesh-Free Hierarchical Clustering methods for Fast Evaluation of Electrostatic Interactions of Point Multipoles. *J. Chem. Phys.* **2017**, *147*, 164104.

(25) Rackers, J. A.; Wang, Z.; Lu, C.; Laury, M. L.; Lagardère, L.; Schnieders, M. J.; Piquemal, J.-P.; Ren, P.; Ponder, J. W. Tinker 8: Software Tools for Molecular Design. *J. Chem. Theory Comput.* **2018**, *14*, 5273−5289.

(26) Case, D. A.; Cheatham, T. E., III.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M., Jr.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber BioMolecular Simulation Programs. *J. Comput. Chem.* **2005**, *26*, 1668−1688.

(27) Todorov, I. T.; Smith, W. *The DL_POLY_4 User Manual*; STFC Daresbury Laboratory: Daresbury, Warrington WA4 4AD, 2014.

(28) Smith, W.; Forester, T. R.; Todorov, I. T. *The DL_POLY Classic User Manual*; STFC Daresbury Laboratory: Daresbury, Warrington WA4 4AD, 2012.

(29) Smith, W.; Forester, T. R. Parallel macromolecular simulations and the replicated data strategy: I. The computation of forces. *Comp. Phys. Commun.* **1994**, *79*, 52−62.

(30) Barnes, J.; Hut, P. A hierarchical O(N log N) force-calculation algorithm. *Nature* **1986**, *324*, 446−449.

(31) Greengard, L. Fast Algorithms for Classical Physics. *Science* **1994**, *265*, 909.

(32) Lindsay, K.; Krasny, R. A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow. *J. Comput. Phys.* **2001**, *172*, 879−907.

(33) Duan, Z.-H.; Krasny, R. An Ewald Summation based multipole method. *J. Chem. Phys.* **2000**, *113*, 3492−3495.

(34) Duan, Z.-H.; Krasny, R. An adaptive treecode for computing nonbonded potential energy in classical molecular systems. *J. Comput. Chem.* **2001**, *22*, 184−195.

(35) Krasny, R.; Wang, L. Fast evaluation of multiquadric RBF sums by a Cartesian treecode. *SIAM J. Sci. Comput.* **2011**, *33*, 2341−2355.

(36) Boateng, H. A.; Krasny, R. Comparison of treecodes for computing electrostatic potentials in charged particle systems with disjoint targets and sources. *J. Comput. Chem.* **2013**, *34*, 2159−2167.

(37) Feng, H.; Barua, A.; Li, S.; Li, X. A parallel adaptive treecode algorithm for evolution of elastically stressed solids. *Commun. Comp. Phys.* **2014**, *15*, 365−387.

(38) Salmon, J. K.; Warren, M. S. Skeletons from the treecode closet. *J. Comput. Phys.* **1994**, *111*, 136−155.

(39) Chen, J.; Wang, L.; Anitescu, M. A fast summation tree code for Matérn Kernel. *SIAM J. Sci. Comput.* **2014**, *36*, A289−A309.

(40) Deng, Q.; Driscoll, T. A. A fast treecode for multiquadric interpolation with varying shape parameters. *SIAM J. Sci. Comput.* **2012**, *34*, A1126−A1140.

(41) Appel, A. W. An efficient program for many-body simulation. *SIAM J. Sci. Stat. Comput.* **1985**, *6*, 85−103.

(42) Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **1987**, *73*, 325−348.

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

(43) Cheng, H.; Greengard, L.; Rokhlin, V. A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.* **1999**, *155*, 468−498.
(44) Giese, T. J.; York, D. M. Extension of Adaptive Tree Code and Fast Multipole Methods to High Angular Momentum Particle Charge Densities. *J. Comput. Chem.* **2008**, *29*, 1895.
(45) Boateng, H. A.; Todorov, I. T. Arbitrary order permanent Cartesian multipolar electrostatic interactions. *J. Chem. Phys.* **2015**, *142*, No. 034117.
(46) Rahman, A.; Stillinger, F. H. Molecular Dynamics Study of Liquid Water. *J. Chem. Phys.* **1971**, *55*, 3336−3359.
(47) Stillinger, F. H.; Rahman, A. Improved simulation of liquid water by molecular dynamics. *J. Chem. Phys.* **1974**, *60*, 1545−1557.
(48) Zhang, C.; Hutter, J.; Sprik, M. Computing the Kirkwood g-factor by combining constant Maxwell electric field and electric displacement simulations: Aapplication to the dielectric constant of liquid water. *J. Phys. Chem. Lett.* **2016**, *7*, 2696−2701.
(49) van der Spoel, D.; van Maaren, P. J.; Berendsen, H. J. C. A systematic study of water models for molecular simulation: Derivation of water models optimized for use with a reaction field. *J. Chem. Phys.* **1998**, *108*, 10220.
(50) Mark, P.; Nilsson, L. Structure and dynamics of liquid water with different long-range interaction truncation and temperature control methods in molecular dynamics simulations. *J. Comput. Chem.* **2002**, *23*, 1211−1219.
(51) Fulton, R. L. The effect of the spatial nonlocality of the Kirkwood g-factor on the determination of the long wavelength dielectric functions in dipolar fluids. *J. Chem. Phys.* **2012**, *136*, 084502.
(52) Weiner, S. J.; Kollman, P. A.; Nguyen, D. T.; Case, D. A. An all atom force-field for simulations of proteins and nucleic-acids. *J. Comput. Chem.* **1986**, *7*, 230−252.
(53) Forester, T. R.; Smith, W.; Clarke, J. H. Molecular Dynamics Simulations of Valinomycin and its Potassium complex in homogeneous solvents. *Biophys. J.* **1996**, *71*, 544−553.
(54) Forester, T. R.; Smith, W.; Clarke, J. H. R. A molecular dynamics study of valinomycin and the potassium-valinomycin complex. *J. Phys. Chem.* **1994**, *98*, 9422−9430.
(55) Forester, T. R.; Smith, W.; Clarke, J. H. R. Capture of potassium ions by by valinomycin: a molecular dynamics simulation study. *J. Phys. Chem.* **1995**, *99*, 14418−14423.
(56) Lee, T.-S.; Cerutti, D. S.; Mermelstein, D.; Lin, C.; LeGrand, S.; Giese, T. J.; Roitberg, A.; Case, D. A.; Walker, R. C.; York, D. M. GPU-Accelerated Molecular Dynamics and Free Energy Methods in Amber 18: Performance Enhancements and New Features. *J. Chem. Inf. Model.* **2018**, *58*, 2043−2050.
(57) Nguyen, T. D.; Carrillo, J.-M. Y.; Dobrynin, A. V.; Brown, W. M. A case study of truncated electrostatics for simulation of polyelectrolyte brushes on GPU accelerators. *J. Chem. Theory Comput.* **2013**, *9*, 73−83.
(58) Brown, W. M.; Kohlmeyer, A.; Plimpton, S. J.; Tharrington, A. N. Implementing molecular dynamics on hybrid high performance computers - Particle-particle particlemesh. *Comput. Phys. Commun.* **2012**, *183*, 449−459.
(59) Brown, W. M.; Wang, P.; Plimpton, S. J.; Tharrington, A. N. Implementing molecular dynamics on hybrid high performance computers - short range forces. *Comput. Phys. Commun.* **2011**, *182*, 898−911.
(60) Bédorf, J.; Gaburov, E.; Zwart, S. P. A sparse octree gravitational N-body code that runs entirely on the GPU processor. *J. Comput. Phys.* **2012**, 2825−2839.
(61) Bédorf, J.; Gaburov, E.; Fujii, M. S.; Nitadori, K.; Ishiyama, T.; Portegies, Z. Pflops on a gravitational tree-code to simulate the milky way galaxy with 18600 GPUs. *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* 2014.
(62) Lukat, G.; Banerjee, R. A GPU accelerated Barnes-Hut tree code for FLASH4. *New Astron.* **2016**, *45*, 14−28.
(63) Miki, Y.; Umemura, M. GOTHIC: Gravitational oct-tree code accelerated by hierarchical time step controlling. *New Astron.* **2017**, *52*, 65−81.

DOI: 10.1021/acs.jctc.9b00648
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX