JCTC Journal of Chemical Theory and Computation

Staggered Mesh Ewald: An Extension of the Smooth Particle-Mesh Ewald Method Adding Great Versatility

David S. Cerutti,[*,†] Robert E. Duke,[‡,§] Thomas A. Darden,[ǁ] and Terry P. Lybrand[†]

Center for Structural Biology, Department of Chemistry, Vanderbilt University, 5142 Medical Research Building III, 465 21st Avenue South, Nashville, Tennessee 37232-8725, Department of Chemistry, University of North Carolina, Campus Box 3290, Chapel Hill, North Carolina 27599-0001, Laboratory of Structural Biology, National Institute of Environmental Health Science, Research Triangle Park, 12 Davis Drive, Chapel Hill, North Carolina 27709-5900, and Open Eye Scientific Software, 9 Bisbee Court, Suite D, Santa Fe, New Mexico 87508-1338

Abstract: We draw on an old technique for improving the accuracy of mesh-based field calculations to extend the popular Smooth Particle Mesh Ewald (SPME) algorithm as the Staggered Mesh Ewald (StME) algorithm. StME improves the accuracy of computed forces by up to 1.2 orders of magnitude and also reduces the drift in system momentum inherent in the SPME method by averaging the results of two separate reciprocal space calculations. StME can use charge mesh spacings roughly 1.5 × larger than SPME to obtain comparable levels of accuracy; the one mesh in an SPME calculation can therefore be replaced with two separate meshes, each less than one-third of the original size. Coarsening the charge mesh can be balanced with reductions in the direct space cutoff to optimize performance: the efficiency of StME rivals or exceeds that of SPME calculations with similarly optimized parameters. StME may also offer advantages for parallel molecular dynamics simulations because it permits the use of coarser meshes without requiring higher orders of charge interpolation and also because the two reciprocal space calculations can be run independently if that is most suitable for the machine architecture. We are planning other improvements to the standard SPME algorithm and anticipate that StME will work synergistically will all of them to dramatically improve the efficiency and parallel scaling of molecular simulations.

1. Introduction

With few exceptions,[1,2] the method of choice for computing long-ranged electrostatic interactions in molecular simulations with periodic boundary conditions is the Ewald sum.[3] Whereas simple truncation of long-ranged electrostatic interactions has been shown to give rise to significant simulation artifacts,[4,5] the use of modern Ewald algorithms enables more efficient simulations with effectively no omission of long-ranged electrostatic interactions.

In its original formulation, the Ewald sum for a system of N particles was an $O(N^2)$ computation, but the introduction of particle:mesh methods have reduced the complexity to $O(N \log N)$[6−8] and even to $O(N)$,[9−11] at which point the choice of optimal algorithm falls to the computational constants of the various methods given the problem's size and particle density. Most concisely, particle:mesh methods rephrase the problem of computing the electrostatic potential of a system of point (or otherwise highly localized) charges from solving $O(N^2)$ pairwise interactions to solving Poisson's equation for a highly smoothed version of the system's

---

* To whom correspondence should be addressed. Phone: (615) 936-3569. Fax: (615) 936-2211. E-mail: david.cerutti@vanderbilt.edu.

† Vanderbilt University.

‡ University of North Carolina.

§ National Institute of Environmental Health Science.

ǁ Open Eye Scientific Software.

Staggered Mesh Ewald: Two Grids Are Better than One

J. Chem. Theory Comput., Vol. 5, No. 9, 2009 **2323**

charges and then determining the difference between this smoothed-charge potential and the system's actual point charge density. This approach can be efficient because the smoothed charge density is written as a Gaussian convolution of the point charge density: the interaction of two Gaussian charges rapidly converges to the interaction of two point charges at distances greater than about six times the Gaussian's root-mean-squared deviation. The mesh-based electrostatic potential can then be solved by fast Fourier transforms (FFTs) in $O(N \log N)$ operations or by a finite-difference Poisson solver in $O(N)$ operations, while the modification needed to recover the point-charge potential is computed in $O(N)$ operations, similar to a simple truncation method.

Molecular simulations require accurate force calculations, as well as values of the total system energy. In the primary publications of the many available electrostatic mesh methods[7,8,10,12] there have been analyses of the parameters such as the width of the Gaussian charge smoothing function, direct space truncation length, and mesh density required to obtain a given degree of accuracy in forces acting on each atom or the total system energy. In simulations, the goal is to balance these parameters to maximize efficiency. Mostly, this is a matter of minimizing the total computational effort, but with the availability of highly scalable molecular dynamics codes,[2,13−15] another critical factor in the computational efficiency of an algorithm is the communications requirement. Parallel implementations on many different machine architectures can therefore benefit from algorithms that can obtain a given level of accuracy with the widest possible range of parameters.

In this Article, we draw upon a technique used in the 1970s for improving the accuracy of force calculations in particle: mesh methods. The method, known then as "interlacing," was first applied to plasma simulations by Chen and colleagues[16] and later to molecular simulations by Eastwood.[17] The fundamental improvement is to use two or more meshes staggered such that their points are displaced by some fraction of the mesh spacing, typically 1/2. Averaging the results obtained from each mesh produces significant error cancellation. When it was introduced, the method was viewed as a means for achieving higher levels of accuracy with limited amounts of computer memory, at the expense of speed. After rediscovering the method, however, we observe that it improves the overall computational efficiency on modern computers and may help to improve the parallel scaling of molecular simulations. We apply interlacing to the popular Smooth Particle Mesh Ewald method, and term the extended method "Staggered Mesh Ewald".

2. Summary of Particle Mesh Ewald Methods

If one calculates the electrostatic potential of a periodic system of charges by applying using Coulomb's law over all pairs of charges in a large number of images of the unit cell, the process is cumbersome and the result is only conditionally convergent. The Ewald method employs a mathematical identity to split the Coulomb sum $E^{(\mathrm{coul})}$ into a "direct space" sum $E^{(\mathrm{dir})}$ that converges rapidly (with a short interparticle distance $|\mathbf{r}_{ij}|$) in real or "direct" space, and a

"reciprocal space" $E^{(\mathrm{rec})}$ sum that converges absolutely in "reciprocal" space after Fourier transformation.

$$
\begin{aligned}
E^{(\mathrm{coul})} &= \frac{1}{2} \sum_{\mathbf{n}} \sum_{i} \sum_{j \neq i} \frac{k_c q_i q_j}{|\mathbf{n} \cdot \mathbf{L} + \mathbf{r}_{ij}|} \\
E^{(\mathrm{dir})} &\simeq \frac{1}{2} \sum_{i} \sum_{j \neq i} \frac{k_c q_i q_j (1 - \mathrm{erf}(\beta |\mathbf{r}|))}{|\mathbf{r}_{ij}|} \\
E^{(\mathrm{rec})} &= \frac{1}{2} \sum_{\mathbf{n}} \sum_{i} \sum_{j \neq i} \frac{k_c q_i q_j \mathrm{erf}(\beta |\mathbf{r}|)}{|\mathbf{n} \cdot \mathbf{L} + \mathbf{r}_{ij}|}
\end{aligned}
\tag{1}
$$

Above, $\mathbf{n}$ represents all unit cell images, including the primary unit cell, $\mathbf{L}$ is a 3×3 matrix whose columns are the unit cell lattice vectors, i and j run over all charged particles in the system, $\mathbf{r}_{ij}$ is the interparticle distance, k_c is Coulomb's constant, and β is the "Ewald coefficient." (Exclusions of electrostatic interactions between bonded atoms in the primary unit cell are omitted from this discussion for simplicity.) The Ewald method reduces the problem of computing the electrostatic energy (and forces on all particles) to an $O(N^2)$ problem, a double sum over all particles to obtain the reciprocal space sum. (Computing $E^{(\mathrm{dir})}$ is an $O(N)$ problem because interactions can be neglected beyond some direct space cutoff L_{cut}.)

Physically, the Ewald method is equivalent to treating the system of point (or otherwise highly localized) charges as a system of diffuse Gaussian charges, solving the electrostatic potential $E^{(\mathrm{rec})}$ and forces $\partial E^{(\mathrm{rec})}/\partial \mathbf{r}_i$ arising from the Gaussian charge density, and then modifying those quantities with $E^{(\mathrm{dir})}$ and $\partial E^{(\mathrm{dir})}/\partial \mathbf{r}_i$ to recover the interactions of the point charges. Ewald mesh methods take this view of the Ewald reciprocal space procedure so that the reciprocal space sum can be solved on a mesh. (The direct space part is identical to the original Ewald method, and will not be discussed further.)

In general, the procedure with any Ewald mesh method entails four stages: (1) interpolate the charge mesh Q given the positions of particles and the magnitudes of partial charges, (2) smooth the interpolated point charges into Gaussian charges of the desired width, (3) compute the electrostatic potential $\Phi^{(\mathrm{rec})}$ by solving Poisson's equation for the smoothed charge density, and (4) compute the electrostatic potential energy and forces given the derivatives of the charge density in Q and the potential $\Phi^{(\mathrm{rec})}$. Many Ewald mesh methods, including the Smooth Particle Mesh Ewald[8] method that we will focus on during the results, make use of Fast Fourier Transforms (FFTs) to solve Poisson's equation; in those cases it is convenient to combine stages 2 and 3. The charge mesh Q is transformed using the forward three-dimensional FFT to obtain $\hat{Q}$, which is then multiplied element-wise by the transformed reciprocal space pair potential $\hat{\theta}^{(\mathrm{rec})}$. The inverse three-dimensional FFT is then applied to the product to complete the convolution $\Phi^{(\mathrm{rec})} = Q \star \theta^{(\mathrm{rec})}$.

The two FFTs needed to convolute Q with $\theta^{(\mathrm{rec})}$ have $O(N \log N)$ computational complexity, much better than the complexity of the original Ewald method. However, for highly parallel molecular dynamics applications, the FFTs still require global data communication: every processor involved in the FFTs must broadcast its part of the problem to all other processors, and in turn receive similar information

Table 1. Test Cases for the Staggered Mesh Ewald Method[a]

case	cell dimensions (a, b, c), Å	cell dimensions (α, β, γ)	atom count
streptavidin	89.7 × 89.7 × 89.7	90°, 90°, 90°	73 305
protein crystal	91.3 × 81.3 × 91.0	90°, 90°, 90°	73 944
glycerol solution	69.7 × 69.7 × 89.0	60°, 90°, 90°	39 808
cyclooxygenase-2	114.8 × 114.8 × 114.8	109.5°, 109.5°, 109.5°	118 833

[a] The cases presented here span a variety of simulation cell geometries. All systems are in the condensed phase and were pre-equilibrated by molecular dynamics simulations at constant pressure.

from every other processor. This constraint on the ultimate scalability of the calculation has driven the development of real-space methods for solving $\Phi^{(rec)}$.[9−11] However, none of these methods has become widely used on commodity hardware because they are all considerably more expensive than the FFT-based methods: the break-even point comes at very high processor counts, which even today are not widely available. The Staggered Mesh Ewald method presented in this communication offers a way to reduce the total amount of mesh data that must be transformed, which we will show can help to accelerate simulations on a single processor and may help extend the scalability of FFT-based Ewald mesh methods.

3. Summary of Mesh Staggering Methods

Mesh staggering, or "interlacing" as it was originally termed, uses multiple samples of the interpolated charge density of particles on a mesh to suppress errors in the mesh calculation because of "aliasing."[7] Interpolation of a particle to a mesh creates a spectrum of aliases for that particle at each mesh point; because the spectrum is not perfectly smooth, the effects of different aliases on other aliases from the same particle or aliases of a nearby particle can be distorted by their proximity on the mesh. The most basic outcome of aliasing is the fluctuation of forces on particles as a function of their alignment relative to the mesh, which in turn is detrimental to momentum and energy conservation. By sampling multiple spectra of each particle on the mesh, different sets of aliases can be generated. Although each of these spectra contains roughly the same level of error in the interactions of each particle's aliases, the errors from multiple spectra may cancel if the spectra evenly sample the possible alignments of the system's particles relative to the mesh. The simplest and most economical implementation of the mesh staggering technique involves mapping particles to two meshes staggered such that points of one mesh fall exactly halfway in between those of the other.[16]

4. Methods

4.1. Preparation of Primary Test Cases. Anticipating that condensed-phase molecular dynamics simulations will be the primary application of the Staggered Mesh Ewald (StME) method, we selected four test systems: a streptavidin tetramer[18] solvated in a cubic cell, a condensed mixture of 35% v/v glycerol and water in a monoclinic cell, a scorpion toxin protein crystal lattice[19] solvated with water and ammonium acetate in an orthorhombic noncubic cell, and a cyclooxygenase-2 (COX-2) dimer[20] solvated in a truncated octahedral cell. Dimensions and atom counts in all of the simulation cells are provided in Table 1. Together, these four

test cases span the available types of periodic simulation cells and encompass a variety of condensed-phase systems.

The SPC/E water model was used in all cases; glycerol parameters were obtained from Chelli and co-workers,[21] and any proteins were modeled with the AMBER FF99SB force field.[22] Prior to electrostatic calculations, all systems were equilibrated with at least 650 ps of molecular dynamics, including position-restrained dynamics if proteins were present and constant-pressure dynamics to reach each system's equilibrium density.

4.2. Accuracy Standards for Ewald Calculations. To compare different Ewald methods, it is necessary to define what parameters determine the accuracy of the calculation and also what is an "acceptable" level of accuracy. We will summarize these parameters here and then assess the efficiency of Ewald methods in terms of acceptable combinations of the parameters in the Results.

The accuracy of the direct space part of any Ewald electrostatics calculation is determined by the direct sum tolerance D_{tol}. Briefly, D_{tol} is the maximum acceptable relative difference between the interaction potential of two Gaussian charges and the interaction potential of two point charges. As we discuss in the Supporting Information, D_{tol} works together with the direct space truncation length L_{cut} to determine the width of the Gaussian charge smoothing function σ.

The accuracy of the reciprocal space part of a Smooth Particle Mesh Ewald (SPME) calculation is primarily a function of the ratio of σ to the mesh spacing μ, but in SPME, there is one other factor involved which is the order of interpolation used to map each point charge to the mesh. For the most generality, we recognize that μ can be different along each of the unit cell dimensions a, b, and c and that the unit cell lattice vectors need not be orthogonal. We therefore discuss results in terms of the number of mesh points in each dimension G_a, G_b, and G_c, in addition to the mesh spacings μ_a, μ_b, and μ_c. Most precisely, the mesh spacings refer to the magnitudes of the bin vectors $\mathbf{v}_a$, $\mathbf{v}_b$, and $\mathbf{v}_c$ as illustrated in Figure 1.

Because there is no single standard for the accuracy of forces in molecular simulations, we chose two based on default SPME parameters from existing molecular dynamics codes. Most codes use the largest μ_a, μ_b, and $\mu_c \leq 1.0$ Å obtainable such that G_a, G_b, and G_c are multiples of 2, 3, and 5; fourth order interpolation (a cubic B-spline) is typically used to map charges to the mesh. The AMBER molecular dynamics modules set $D_{tol} = 1.0 \times 10^{-5}$ and $L_{cut} = 8.0$ Å by default, whereas values of $D_{tol} = 1.0 \times 10^{-6}$ and $L_{cut} = 12.0$ Å are recommended in the NAMD and CHARMM communities. Because the overall strength of atomic charges differs between systems and slight changes
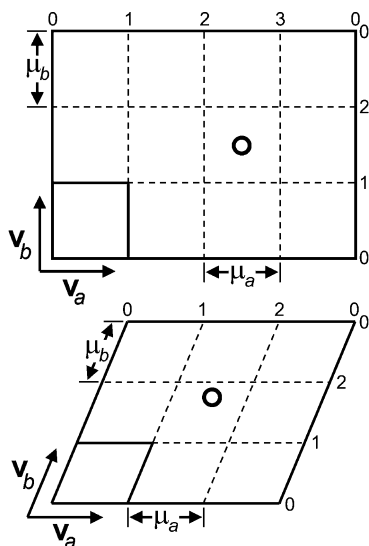
Staggered Mesh Ewald: Two Grids Are Better than One

J. Chem. Theory Comput., Vol. 5, No. 9, 2009 **2325**

Figure 1. Graphical guide to mesh terminology used in the text. Two examples of a two-dimensional mesh are given. The upper mesh is a rectangular mesh analogous to an orthorhombic three-dimensional unit cell; the lower mesh is analogous to a nonorthorhombic unit cell. We define the bin vectors, $\mathbf{v}_a$ and $\mathbf{v}_b$, as shown for each mesh; note that the magnitudes of the bin vectors correspond to the mesh spacings μ_a and μ_b and that the lattice vectors can be written as $G_a\mathbf{v}_a$ and $G_b\mathbf{v}_b$, where G_a and G_b are the number of mesh cells in each dimension. Each mesh point is indexed from 0 to $G_a - 1$ or $G_b - 1$ and the meshes span a periodic unit cell as shown on the diagram. The mesh bin coordinates, $\mathbf{u} = (u_a, u_b)$ describe the location of a point $\mathbf{r}$ within the mesh: $\mathbf{r} = u_a\mathbf{v}_a + u_b\mathbf{v}_b$. Each mesh contains a small circle to represent a particle; the expression for its bin displacement is given by eq 2 or $\xi = \mathbf{u} - \text{floor}(\mathbf{u})$. In the upper mesh, the particle has mesh bin coordinates $\mathbf{u} = (2.5, 1.5)$ and bin displacements $\vec{\xi} = (0.5, 0.5)$; in the lower mesh, the particle's mesh bin coordinates are $(1.5, 1.75)$, and its bin displacements are $(0.5, 0.75)$.

in the size of each system may graduate G_a, G_b, or G_c to the next available integer (i.e., 80 to 90 or 108 to 120), the accuracy of either method is system-specific. We therefore computed forces on all atoms from the four test cases in Table 1 using each set of Ewald parameters and compared them to the results of regular Ewald calculations as described above. We concluded that the AMBER default Ewald parameters can be expected to yield forces accurate to within 7.5×10^{-3} kcal/mol Å, roughly 0.05% relative error, whereas those recommended by the NAMD and CHARMM communities yield forces roughly five times more accurate, to within 1.5×10^{-3} kcal/mol Å or 0.01% relative error. We will refer to these as the "AMBER" and "CHARMM" standards later in this work.

In defining these standards, we emphasize that the default *settings* of a particular molecular dynamics package are separate from the numerical stability of the code itself. The AMBER dynamics engines SANDER and PMEMD both use double precision for all computations and can run simulations with very little energy drift. We also emphasize that the level of accuracy *necessary* to obtain reliable simulation results is not precisely known. The

"AMBER" and "CHARMM" standards merely represent two points on a continuum.

4.3. Smooth Particle Mesh Ewald Force Calculations. SPME calculations for this work were performed using the SANDER module of the AMBER software package[13] in debugging mode to print out the forces. Staggered Mesh Ewald calculations, presented in the results, were done by averaging the results of two SPME calculations using the appropriate alignments of the particles and mesh. A high-accuracy regular Ewald sum, in which the forces were converged to a precision of 1.0×10^{-5} kcal/mol Å, was used as the reference for rating the accuracy of any Ewald mesh calculation.

5. Results

Although mesh staggering has been applied to the Particle: Particle Particle:Mesh (P[3]M) method,[17] we will first quantify its benefits in the context of the newer Smooth Particle Mesh Ewald (SPME) method for simple cases before moving on to complex molecular systems. We divide the numerical error caused by particle aliasing into two sources: self-image forces that particles exert on themselves and errors in pair interaction forces. As would be expected, we observed that the self-image force errors are proportional to the squares of the individual charges and that pair interaction force errors are proportional to the product of the two charges. However, to simplify the following presentation, we use only $+1e$ and $-1e$ charges, where e is the charge of a proton. We also emphasize that the interpolation order, L_{cut}, and D_{tol} significantly influence the accuracy of SPME calculations, but again to keep the presentation simple, we fix these parameters at $L_{\text{cut}} = 9.0$ Å, $D_{\text{tol}} = 1.0 \times 10^{-6}$, and fourth-order interpolation. The periodic unit cell in the following examples, termed the "test cell," was a 64 Å cube.

5.1. Self-Image Forces in Smooth Particle Mesh Ewald calculations. The first source of numerical error can be observed by computing the SPME force on a single particle. We placed a single charge at a random point in the test cell and computed the electrostatic forces on the particle using a mesh spacing μ of 1.333 or 1.000 Å (corresponding to $G = 48$ or 64 points on a side). Repeating this procedure many times allowed us to plot the error in the force on the particle as a function of its alignment on the mesh, which we describe as its bin displacement ξ

$$\xi_\alpha = \frac{\mathbf{r}_\alpha}{\mu_\alpha} - \text{floor}\left(\frac{\mathbf{r}_\alpha}{\mu_\alpha}\right) \qquad (2)$$

where $\mathbf{r}$ is a particle's position relative to the origin of the grid and $\alpha \in \{a, b, c\}$, the three dimensions of the mesh. The concept of a bin displacement is illustrated in Figure 1.

Simply stated, the self-image force error is any deviation from zero as a charge should put no force on itself and forces resulting from the charge's images should perfectly cancel. In Figure 2, the three components of the error are shown to be separable in three dimensions, plotted against the corresponding values of ξ. As has been found with previous investigations on mesh staggering, the self-image force error $\mathbf{F}^{(\text{si})}$ is well described by a Fourier sine series (eq 3)

$$\mathbf{F}_\alpha^{(si)} \simeq \sum_{p=0}^{\infty} W_\alpha^{(p)} q^2 \sin(2p\pi\xi_\alpha) \tag{3}$$

where q is the atomic charge and the $W^{(p)}$ coefficients depend on the direct space cutoff, mesh spacing in each dimension, and the interpolation order. For the cubic test cell, $W_a^{(p)} = W_b^{(p)} = W_c^{(p)}$. Repeating the mesh calculation for a mesh staggered by $\mu/2$ in all dimensions would eliminate all errors associated with sine series terms with odd values of k, most importantly $k = 1$. However, eq 3 raises the possibility of eliminating self-image force errors for all values of k by simply computing the appropriate sine series coefficients.

To see how $\mathbf{F}^{(si)}$ contributes to the total error in a system of multiple charges, we created a sparse set of 200 $+1e$ and 200 $-1e$ charges spaced further than $L_{cut} = 9.0$ Å from one another in the test cell. SPME calculations were carried out as before and were compared to the results from a regular Ewald calculation as described in Methods. The results in Figure 3 show that $\mathbf{F}^{(si)}$ plays a major role in the total error of the SPME calculation for a system of sparse charges; for all particles, the numerical error correlates with $\mathbf{F}^{(si)}$ with Pearson coefficient 0.96 ± 0.01 for $\mu = 1.000$ or 1.333 Å; if we optimize the first two $W^{(p)}$ coefficients to reduce the error in these SPME calculations, the values come very close to those found for the case of solitary charges under similar conditions, as illustrated in Figure 3. Correcting for the self-image forces can improve the accuracy of either of these SPME calculations by a factor of 4 to 5, implying that for simulations of diffuse plasmas the benefits of a second, staggered mesh calculation can be obtained by simply computing the appropriate sine series coefficients and applying a correction force to each particle after each mesh calculation. However, there are clearly other sources of error even with the particles spaced by more than L_{cut}.

As shown in Figure 4, these other sources of error dominate in a condensed system. While the overall error remains weakly correlated with $\mathbf{F}^{(si)}$, we found that simply correcting the self-image error was no longer effective for improving the accuracy of SPME calculations on dense plasmas or solvated biomolecular systems.

5.2. Pair Interaction Force Errors in Smooth Particle Mesh Ewald Calculations. To analyze the pair interactions that seem to be critical for accurate SPME calculations in condensed-phase systems, we set two charges of opposite sign close to one another in the test cell and computed the force between them using the same SPME parameters as before. We then iteratively perturbed the second charge along the x axis and recomputed the forces until the charge had traveled the entire width of the test cell. By repeating this analysis for different fixed positions of the first charge relative to the mesh (sampling the bin displacement ξ for the first charge while sampling all possible x coordinates of the second charge), we were able to plot the error in pair interaction forces between two particles as shown in Figure 5.

As shown in Figure 5, different aliases of each particle interact in complex ways, giving rise to errors that depend both on the interparticle separation in all three dimensions as well as the bin displacement of the first charge. Despite these complexities, however, Figure 5 confirms that mesh staggering eliminates a majority of the pair interaction force errors, particularly if the two meshes are staggered by 1/2 the mesh spacing μ in all directions simultaneously. Chen and co-workers used this multidimensional staggering approach in their simulations of plasmas,[16] although other investigators[7] have suggested that the results of as many as eight meshes, staggered by $\mu/2$ along any and all of the unit cell lattice vectors, should be averaged to obtain the best results. We tried averaging the results of eight such meshes (data not shown), but found this much more expensive approach to give scarcely better results than using only two meshes.
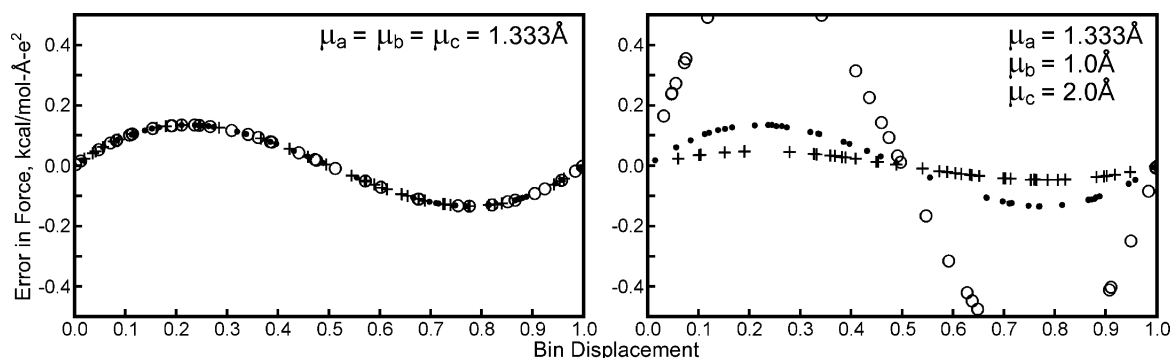

Figure 2. Force errors associated with mapping a single atomic charge to the mesh. When point charges are mapped to a mesh, they suffer an artifactual net force from their own the self-images in other unit cells. These artifactual forces decrease rapidly as the mesh becomes finer but can be significant even for 1.0 Å meshes, a common spacing used in conjunction with direct space cutoffs of ∼9 Å. The plots above show the self-image forces on a $+1e$ test charge as it is moved to many random positions inside a 64 Å cubic box, when the mesh spacings given in each panel are used to compute the reciprocal space electrostatics. ●, +, and ○ represent self-image forces in the x, y, and z directions (along $\mathbf{v}_a$, $\mathbf{v}_b$, and $\mathbf{v}_c$ for this mesh). In all cases the self-image forces have a sinusoidal form given in eq 3 with respect to the bin displacement; the amplitude of the error increases rapidly with the mesh spacings μ_a, μ_b, or μ_c but appears to depend only on the charge's bin displacement in each dimension. Molecular dynamics codes typically add a "net force correction" to prevent the reciprocal space calculation from imparting artificial momentum on the system; eliminating the self-image forces would reduce but not obviate the need for such a correction (see also Figure 3).
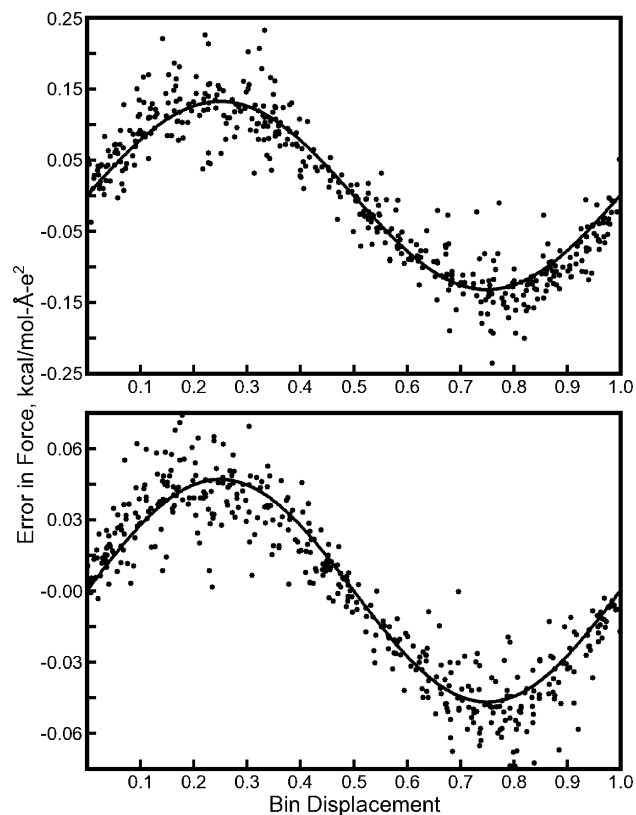
Staggered Mesh Ewald: Two Grids Are Better than One

J. Chem. Theory Comput., Vol. 5, No. 9, 2009 **2327**

Figure 3. Force errors in a system of many sparse charges. A total of 200 pairs of $\pm 1e$ charges were placed in another 64 Å cubic box similar to the setup in Figure 2. Charges were distributed such that no two came within 9.5 Å of one another (for this example, $L_{cut} = 9.0$ Å, and $D_{tol} = 1.0 \times 10^{-6}$). Black dots in each panel represent the total error in the force on each charge in the *x*, *y*, or *z* directions. The black lines in each panel represent the expected self-image forces, obtained by optimizing the coefficients $W^{(1)}$ and $W^{(2)}$ in eq 3 for each mesh. For $\mu = 1.333$ Å, $W^{(1)} = 0.1335$ and $W^{(2)} = 0.0112$; for $\mu = 1.0$ Å, $W^{(1)} = 0.0476$ and $W^{(2)} = 0.0046$. While the expected self-image forces account for a significant amount of the total error, other sources of error are clearly present even for this sparse system of charges.

5.3. Staggered Mesh Ewald Method. Having confirmed that mesh staggering can eliminate large portions of the self-interaction force error, as well as pair interaction force errors in the SPME method, we sought to quantify the benefits of the mesh staggering in terms of accuracy and overall calculation efficiency when applied to condensed-phase biomolecular systems.

We term the use of two reciprocal space calculations on meshes aligned one-half mesh spacing relative to one another in all three mesh dimensions "Staggered Mesh Ewald" (StME). Because the reciprocal space operations (mapping charges to the mesh, convoluting the density and solving Poisson's equation, and interpolating forces from the smoothed potential) are identical to the procedures in SPME, implementing this method in current molecular dynamics codes can be straightforward. However, we will suggest some additional optimizations later in the Results.

With two meshes to compute but the potential to increase the accuracy by an order of magnitude or more relative to the corresponding SPME calculation, we wanted to thor-
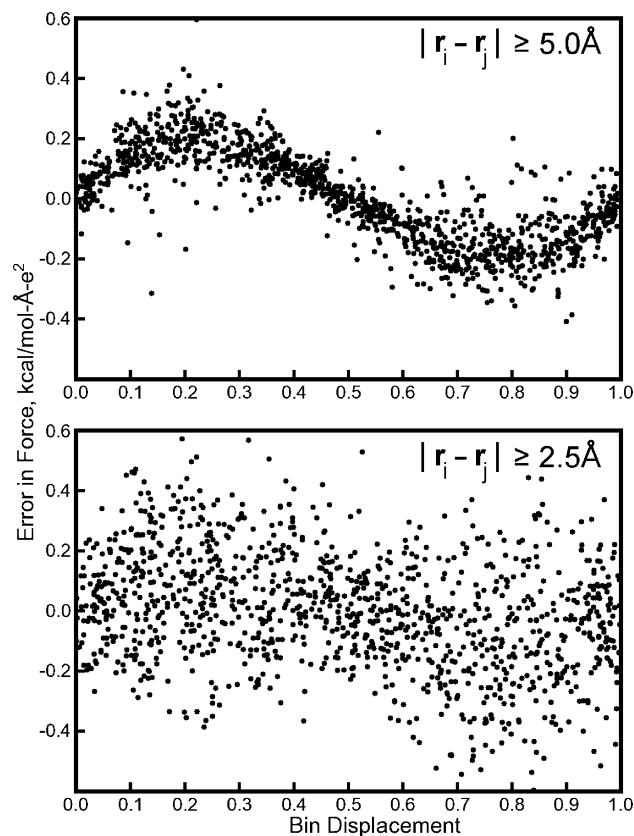

Figure 4. Force errors in increasingly dense systems. Many pairs of $\pm 1e$ charges (1600 in the top panel, 12 800 in the bottom panel) were placed in a 64 Å cubic box in the same manner as in Figure 3 according to the minimum interparticle spacing $|\mathbf{r}_i - \mathbf{r}_j|$ given in each panel. For this example, $\mu = 1.333$ Å, $L_{cut} = 9.0$ Å, and $D_{tol} = 1.0 \times 10^{-6}$. Black dots again represent the total error in the force on each charge in the *x*, *y*, or *z* directions (errors for only 400 charges are shown for clarity). The self-image forces remain a major factor in the total error even at minimum interparticle spacings as low as 5.0 Å, but other sources of error rapidly dominate as the minimum spacing goes below 2.5 Å and thus removing the self-image error is no longer an effective correction for coarse reciprocal space meshes. In a typical MD simulation, interparticle separations of less than 1 Å are common.

oughly characterize the numerical error of StME relative to SPME for a variety of simulation parameters. In the 1970s, FFT solvers were efficient with mesh sizes of powers of 2, at the time, "interlacing" typically meant using two coarse meshes with twice the spacing of the equivalent fine mesh, and delivered an intermediate level of accuracy. Modern FFT solvers, however, are able to work efficiently with multiples of 2, 3, 5, and even 7; we therefore have much more freedom in the choice of mesh spacings for maximizing efficiency. Furthermore, since the 1970s, simulations in nonorthorhombic unit cells have become more common; it is important to confirm that mesh staggering is beneficial in these cases as well.

We performed both SPME and StME calculations on all test cases listed in Table 1 and compared them to regular Ewald calculations as described in Methods. The results are plotted in Figures 6 and 7. These tests, which included noncubic and nonorthorhombic cells, demonstrate
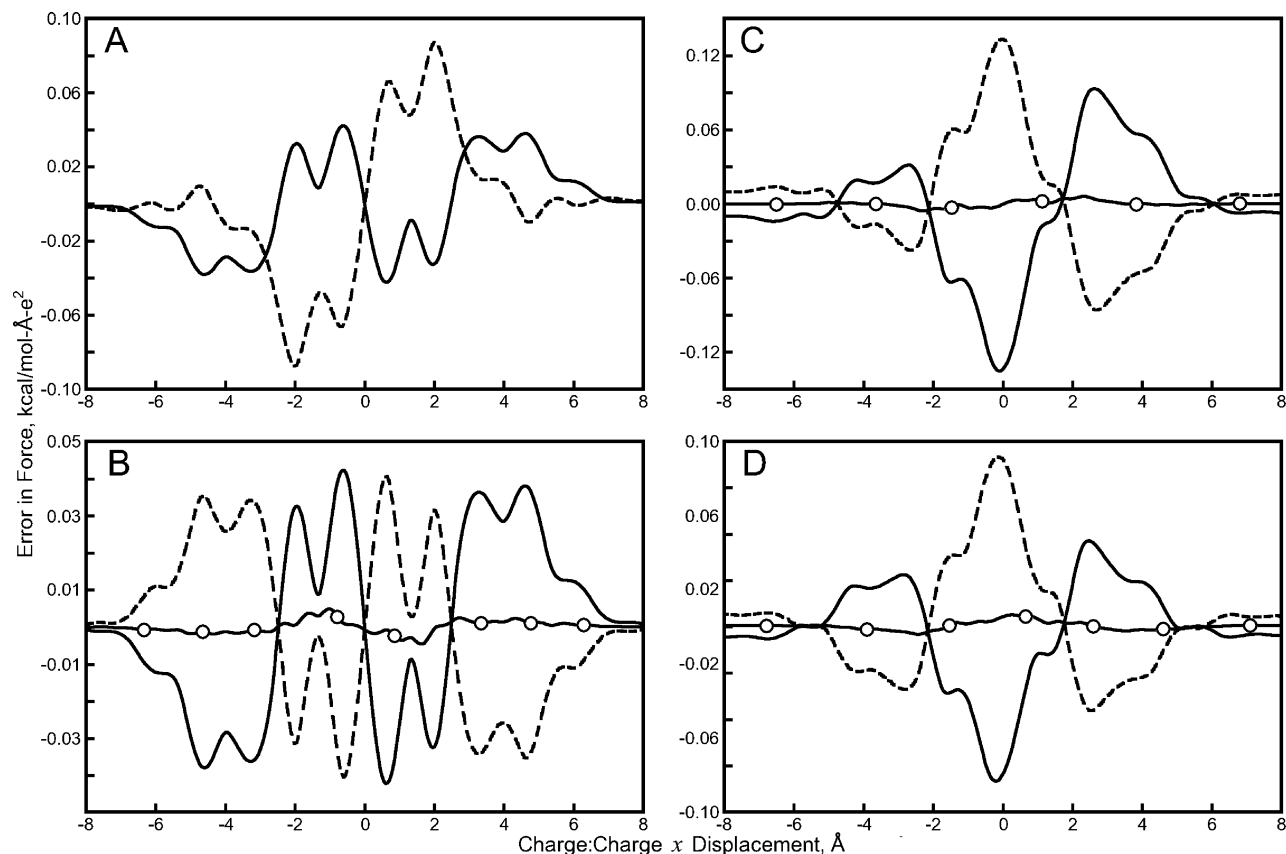
Figure 5. Force errors associated with pair interactions caused by coarse reciprocal space meshes. Significant errors enter the calculation of the force between two $\pm 1e$ charges P_1 and P_2 when a coarse mesh (in this case, $\mu = 1.333$ Å) is used. Error in the x component of the force exerted on P_1 by P_2, excluding any self-image force, is plotted as P_2 is moved parallel to the x axis such that its path intercepts P_1 unless otherwise noted. Results for several different positions of P_1 are shown as a function of the x displacement between P_1 and P_2. Panel A, solid line: P_1 is positioned at the origin. Panel A, dashed line: P_1 is positioned at 0.5μ along the x axis. In Panel B, the solid line is copied from Panel A, but for the dashed line, P_1 is positioned at $(0.5 \mu, 0.5 \mu, 0.5 \mu)$. The errors are anticorrelated in Panel A and more strongly so in Panel B (the solid line with circles shows the average of the two errors). Panels C and D follow the format of Panel B. Panel C, solid line: P_1 positioned at 0.25μ on the x axis. Panel C, dashed line: P_1 positioned at $(0.75 \mu, 0.5 \mu, 0.5 \mu)$. Panel D, solid line: P_1 positioned at $(0.307 \mu, 1.421 \mu, 1.804 \mu)$, P_2 moved along the x axis. Panel D, dashed line: P_1 positioned at $(0.807 \mu, 1.921 \mu, 2.804 \mu)$, P_2 moved to sample points $(x, 0.5 \mu, 0.5 \mu)$.

the applicability of the method to periodic systems in general. Note, however, that the mesh grids are staggered by $(1/2)(\mathbf{v}_a + \mathbf{v}_b + \mathbf{v}_c)$, not simply by the half the mesh spacing in x, y, and z.

Figures 6 and 7 show that, for a given order of interpolation and direct sum tolerance, the StME method can greatly increase the accuracy of computed forces relative to an SPME calculation with similar parameters. In all systems, error reductions exceeding 1 order of magnitude can be obtained for direct space cutoffs of 8 Å to 10 Å with similar mesh sizes and values of D_{tol}; the benefits of StME appear to be highest for direct space cutoffs and mesh densities near those used in typical MD simulations. Furthermore, the StME method continues to produce comparable increases in accuracy, relative to SPME, if the order of interpolation is increased or if D_{tol} is reduced. Although we did not demonstrate that the self-image forces and errors in pair interaction forces could be canceled with staggered meshes in nonorthorhombic unit cells, StME shows equally good performance for other condensed-phase systems in such unit cells.

While electrostatic potentials from two meshes must be computed in StME, the method achieves comparable accuracy to SPME with coarser meshes and smaller values of L_{cut}. For example, when using fourth order interpolation and $D_{tol} = 1.0 \times 10^{-5}$ for calculations on the streptavidin system, StME achieves nearly the same accuracy with $L_{cut} = 8.0$ Å and $G_a = G_b = G_c = 60$ as SPME with $L_{cut} = 9.0$ Å and $G_a = G_b = G_c = 90$. In such a case, the direct space workload is reduced by almost 30% and the overall FFT workload is reduced by more than 40%: each mesh of 60^3 points is 3.375 × smaller than the mesh of 90^3 points that would become the bottleneck for the SPME calculation. Twice as many charge mapping and force interpolations would be required in the most basic implementation of StME, but as will be discussed, other optimizations can still lead to significant improvements in simulation efficiency with this method. We will make a detailed analysis of the optimal parameters for StME and SPME calculations later in the results.

5.4. Energies and Virials Obtained with Staggered Mesh Ewald. Highly accurate forces are the most important product of a molecular dynamics method, but we also wanted

Staggered Mesh Ewald: Two Grids Are Better than One

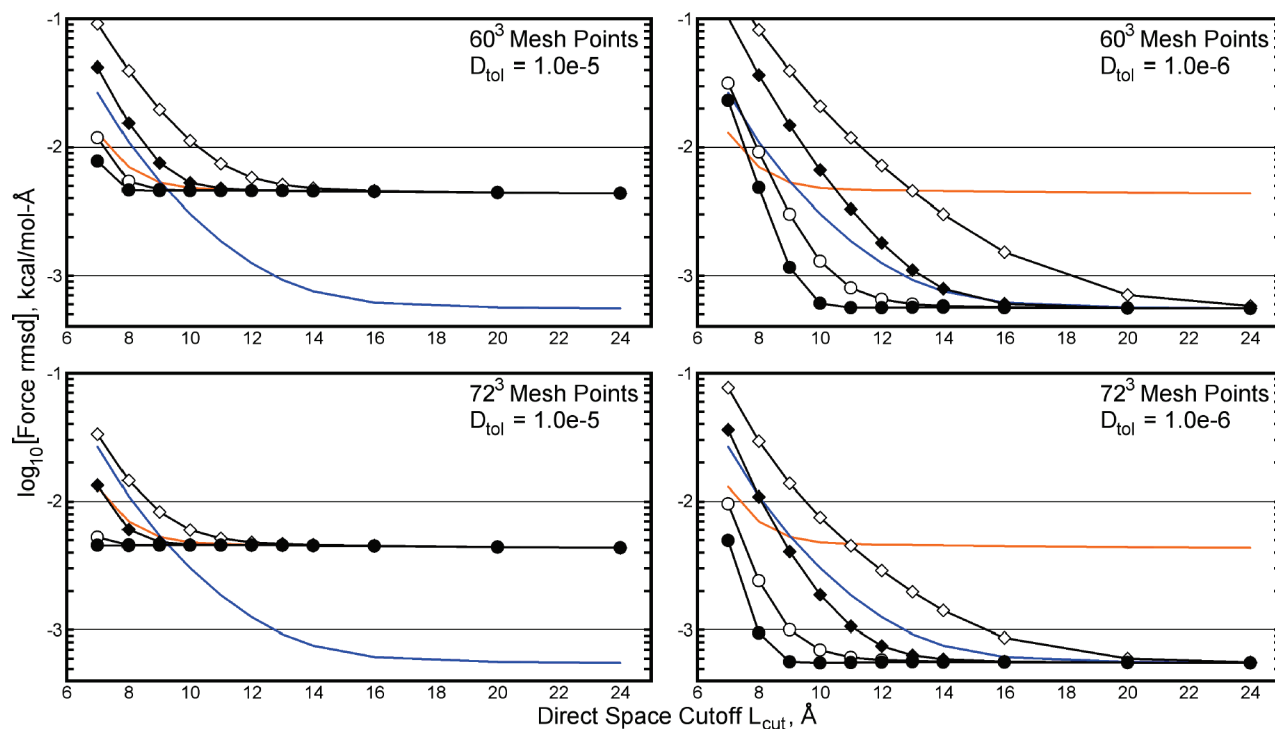*J. Chem. Theory Comput., Vol. 5, No. 9, 2009* **2329**

Figure 6. Accuracy of Smooth Particle Mesh Ewald (SPME) and Staggered Mesh Ewald (StME) calculations for the streptavidin test case. Each type of Ewald calculation was run using the parameters given in the top right corner of each panel. Black lines with open or filled symbols represent 4th or 5th order interpolation, respectively; diamonds and circles represent SPME and StME calculations, respectively. In most MD codes, a mesh of 90^3 points would be used, along with $D_{tol} = 1.0 \times 10^{-5}$ or 1.0×10^{-6} and 4th order interpolation; these cases are shown in orange and blue, respectively, for reference. Additional details of the streptavidin system are given in Table 1. Even with a mesh spacing $1.5\times$ the standard value, the StME method offers improved accuracy over the corresponding SPME calculation for nearly all values of L_{cut}. StME maintains its advantage with finer meshes or higher interpolation orders.

to test whether StME could produce energies and virials of comparable accuracy to SPME, particularly when used with coarser meshes. Typically, electrostatics dominates the total potential energy of a molecular system but the energy of the reciprocal space part is fairly small. The reciprocal space calculation also makes fairly minor contributions to the system's virial tensor. Still, errors in these contributions could limit the overall applicability of StME. We also tested whether errors in either of these quantities were systematic or random by repeating the StME and SPME calculations for 20 individual snapshots taken at 100 ps intervals from 2 ns trajectories of each system. These tests were conducted using the SANDER module of the AMBER software.

When the mesh used to compute the reciprocal space electrostatic potential is coarsened, both the energy and elements of the virial tensor trace become increased relative to their values obtained with a very fine mesh. Figure 8 shows that for a mesh spacing μ approaching 1.5 Å, the reciprocal space calculation begins to report energies noticeably different from the values obtained in the limit of a very fine mesh. This behavior holds for the reciprocal space contributions to the virial trace as well; the off-diagonal elements of the virial accumulate very large errors (data not shown).

Despite these limitations, it appears that StME can hold its ground in most constant pressure simulations and in cases when the total system energy is required. In any periodic simulation cell, isotropic position rescaling can be used to

adjust the cell size to satisfy a particular external pressure making use of only the trace of the virial tensor. In orthorhombic cells, even anisotropic rescaling can be accomplished without reference to the virial's off-diagonal elements. In our test cases, elements of the virial's trace are consistently biased by nearly the same amount for a given system and a particular set of SPME parameters.

In StME calculations, the strong anticorrelation between the errors of the two coarse meshes carries over into estimates of the energy and virial trace, so that the average of the two results is biased more consistently than either alone. It is likely that the necessary correction factors can be computed at the beginning of a simulation by comparing the results from both StME meshes to the results from a finer mesh or higher interpolation order computed with the same values of D_{tol} and L_{cut}. Periodic updates of the correction factors, perhaps every 10 000 steps or upon significant changes in the unit cell dimensions, appear to be a reliable means of keeping errors in the calculated energy and virial trace within the levels obtained with conventional SPME calculations and accepted parameters. To ensure that the accuracy in estimates of the energy and virial trace (after removing the bias) was comparable to the accuracy of forces in StME, we scanned over a large number of all four Ewald parameters for the streptavidin and COX-2 test cases. The results in Figure 9 confirm that, if StME produces accurate forces, it produces a precise virial trace and energy as well. Further explanation of why the energy and
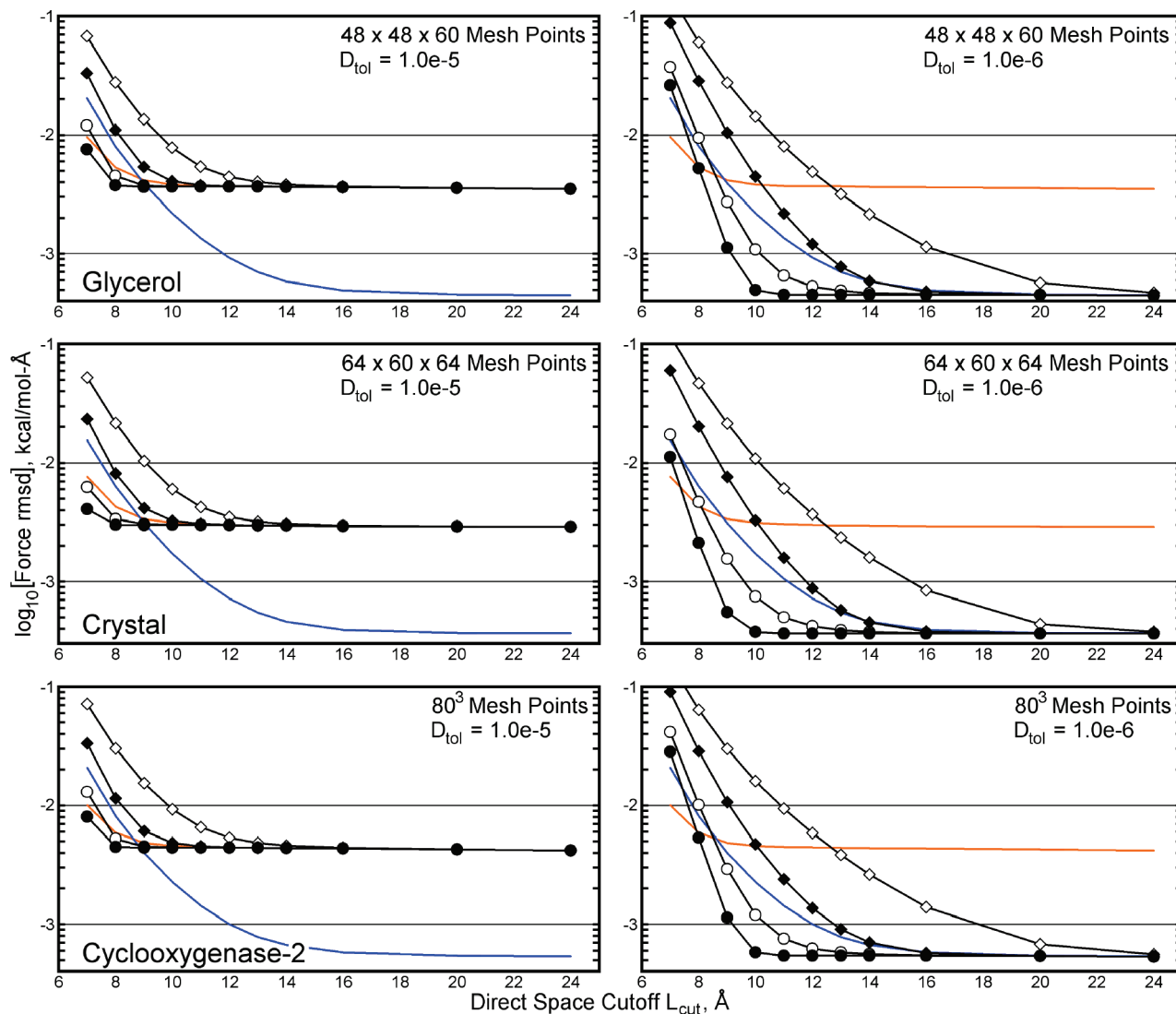
Figure 7. Accuracy of SPME and StME calculations for three other test cases. The format follows Figure 6 but only the case of a mesh spacing 1.5× the default value, that is μ approaching 1.5 Å, is shown. The 35% v/v glycerol:water mixture, protein lattice, and the solvated COX-2 dimer are simulated in monoclinic, orthorhombic, and truncated octahedral cells, respectively. For efficiency, the truncated octahedron is tiled and reshaped into a triclinic unit cell in dynamics simulations. Additional details of all systems can be found in Table 1. The StME method shows comparable performance relative to SPME across all test cases.

virial trace estimates are biased in the manner observed is provided in the Supporting Information.

5.5. Simple Metric for the Accuracy of Ewald Mesh Methods. As in shown in the Supporting Information, it is logical to compute the accuracy of the reciprocal space part of an Ewald mesh calculation as a function of σ/μ, where σ is the width of the Gaussian charge smoothing function defined in Equation S.1 of the Supporting Information and μ, again, is the mesh spacing. We did this for all four of our test cases by computing SPME or StME calculations for σ ranging from 0.5 to 3.0 Å, μ ranging from 0.9 to 2.0 Å, and fourth, fifth, or sixth order interpolation.

Force errors for each test case are plotted as a function of σ/μ in Figure 10 for StME calculations using fourth order interpolation and SPME calculations using fourth, fifth, or sixth order interpolation. In all cases, the accuracy of forces appears to approach a log-linear relationship with σ/μ; in

this region, the accuracy of StME is roughly 1.2 orders of magnitude higher than SPME with identical parameters.

Because the accuracy in an Ewald mesh calculation also depends on contributions from the direct space calculation, we assumed that the entire electrostatics calculation could meet "AMBER" or "CHARMM" accuracy if the reciprocal space calculation produced errors up to half the level of either standard (this estimate is conservative, as the direct and reciprocal space forces for any given atom are generally oriented randomly with respect to one another, so the magnitude of the combined error will be at most the sum of the direct and reciprocal space errors). Our findings echo results in Figures 6 and 7: StME calculations using fourth order interpolation can meet the "AMBER" level of accuracy with $\sigma \geq 1.0 \ \mu$, whereas SPME run with fourth order interpolation would require $\sigma \geq 1.5 \ \mu$. This is reflected in the AMBER default parameters: $L_{cut} = 8.0$ Å and $D_{tol} = 1.0$
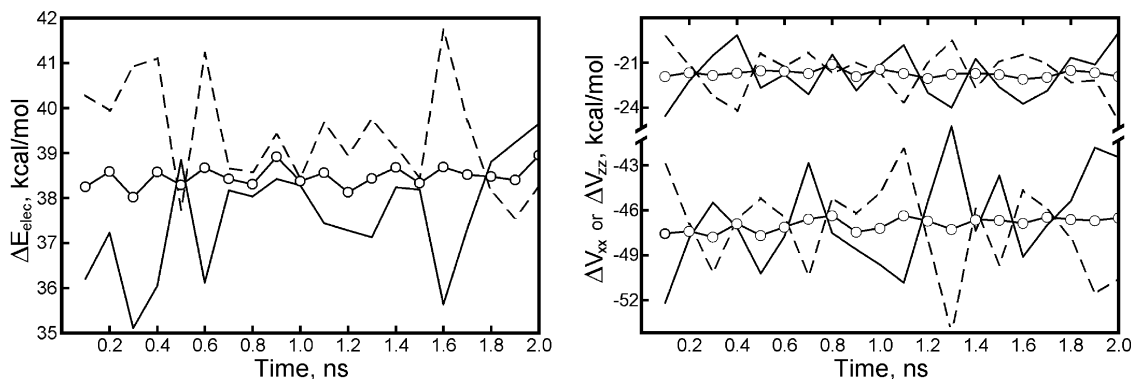
Figure 8. Removable bias in StME estimates of electrostatic energy and elements of the virial tensor trace. In the left panel, StME using 60^3 mesh points, 4th-order interpolation, $D_{tol} = 1.0 \times 10^{-5}$, and $L_{cut} = 8.0$ Å estimates the streptavidin test system's electrostatic energy 38.5 kcal/mol too high relative to a very accurate standard, on average, over the course of a 2 ns simulation. The StME error in the energy estimate is given by the solid line with open circles; plain solid and dashed lines show the error if either of the two StME meshes were used alone. In comparison, a standard SPME calculation run with the AMBER default parameters (a mesh of 90^3 points) delivers an error of only 0.6 ± 0.1 kcal/mol. However, the errors in the reciprocal space energy estimates of each StME mesh are strongly anticorrelated such that the overall error is very consistent: 38.5 ± 0.2 kcal/mol. If the 38.5 kcal/mol bias is removed by measuring against a high accuracy standard occasionally over the course of a simulation, the electrostatic energy can be consistently estimated to within the error of the AMBER default parameters. A similar treatment can be applied to derive the correct reciprocal space virial tensor trace, as shown for the COX-2 test case in the right panel (StME was performed with meshes of 80^3 points; the default AMBER parameters imply a mesh of 120^3 points for SPME). The format of the lines is the same; error in V_{xx} appears in the bottom half of plot, error in V_{zz} in the top half (error in V_{yy} is omitted for clarity).

$\times 10^{-5}$ implies that σ is approximately 1.42 Å to pair with $\mu \leq 1$ Å. Similarly, StME can achieve "CHARMM" accuracy using fourth order interpolation and $\sigma \geq 1.2 \mu$. SPME with fourth order interpolation would require $\sigma \geq 2.0 \mu$: this is reflected by the NAMD recommended parameters $L_{cut} = 12.0$ Å and $D_{tol} = 1.0 \times 10^{-6}$, implying $\sigma \approx 1.94$ Å for $\mu \leq 1$ Å. Figure 10 also shows that SPME calculations must use sixth-order interpolation to produce "AMBER" or "CHARMM" accuracy with the σ/μ ratios available to StME with fourth order interpolation.

In the Supporting Information, we show that there is a nearly linear relationship between σ and L_{cut} for a given D_{tol}, which implies that there is then a roughly linear relationship between the acceptable values of L_{cut} and μ for a given value of D_{tol}. In the next section, we will examine how the combination of μ, L_{cut}, and D_{tol} can be used to optimize performance in both SPME and StME.

5.6. Optimal StME and SPME Parameters. To define optimal Ewald parameters, we return to the "AMBER" and "CHARMM" accuracy standards as defined in Methods. Because different computing architectures favor different levels of real-space or reciprocal space calculations, we did intensive scans of L_{cut} between 6.0 and 16.0 Å, μ between 0.7 and 2.0 Å, and D_{tol} between 5.0×10^{-7} and 1.0×10^{-5} for both fourth and fifth order interpolation. Because the choice of D_{tol} does not affect execution time, we sought any value of D_{tol} that could satisfy the accuracy standards for given values of L_{cut} and μ. Results for SPME and StME methods are plotted in Figure 11.

While increasing the interpolation order from 4 to 5 will greatly expand the combinations of L_{cut} and μ that can produce a particular level of accuracy, staggered meshes with fourth order interpolation offer an even wider array of options. Figure 11 also confirms a result evident in Figures

6 and 7 that increasing the interpolation order with staggered meshes is of marginal benefit when seeking the AMBER level of accuracy but offers more significant improvements when seeking the higher CHARMM level of accuracy.

Although we do not have a working version of Staggered Mesh Ewald in an efficient molecular dynamics package, it is not difficult to obtain reasonable estimates of the single-processor efficiency of StME versus SPME. We assume that the costs of the occasional energy and virial bias corrections and the cost of averaging the forces obtained by the two reciprocal space calculations are negligible. If the two reciprocal space calculations share data, the computation of the reciprocal space pair potential $\hat{\theta}^{(rec)}$ need only be done once for both meshes, and there is even the possibility of using "harmonic averaging,"[17] combining the two staggered meshes in Fourier space to eliminate one of the four FFTs and one of the two force interpolation procedures for significant overall savings. However, we assumed that the two calculations must be done independently, because there may be benefits to parallel performance in this regard and independent calculations make the StME implementation trivial. Under these assumptions, the reciprocal space part of an StME calculation takes exactly twice as long as the identical SPME reciprocal space calculation. Tests were conducted on an Intel 2.66 GHz E5430 processor with the serial version of the pmemd module of AMBER10. Similar to findings presented by Crocker and co-workers in the development of their own parameter optimization program MDSimAid,[23] we were unable to significantly improve the performance of single-processor SPME calculations by simply adjusting the parameters. However, Tables 2 and 3, which also provide additional details of the molecular dynamics benchmark, shows that optimized StME parameters can perform somewhat better than optimized SPME param-
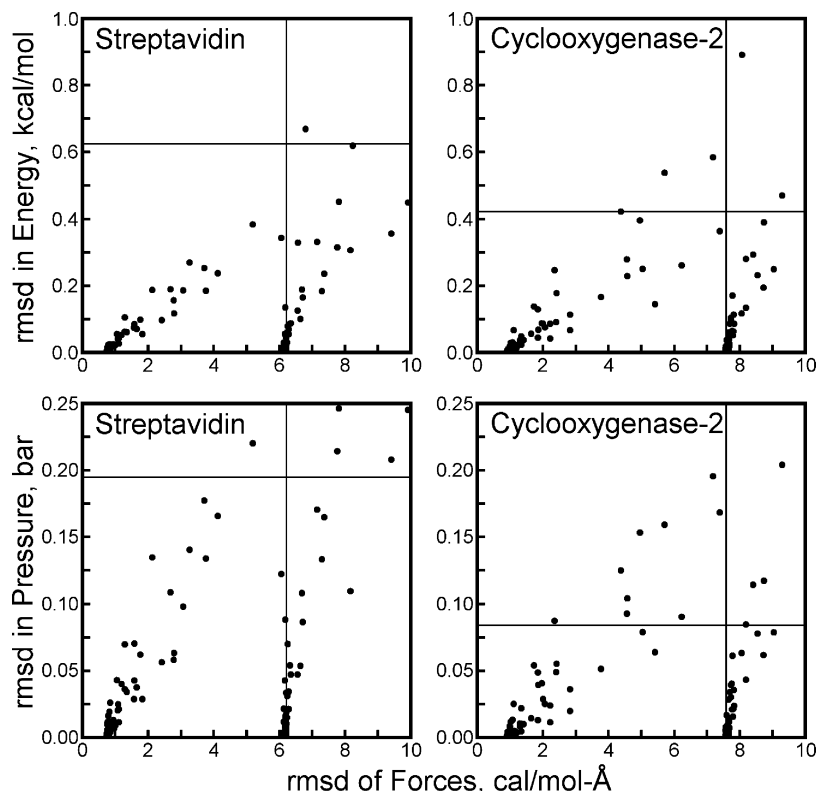
Figure 9. Error in energy and virial trace elements plotted against error in forces obtained by the StME method. Forces, energies, and virials were computed by StME for 20 snapshots of the streptavidin and COX-2 test cases for D_{tol} of 1.0×10^{-5} or 1.0×10^{-6}, μ ranging from 2.0 to 0.9 Å, L_{cut} ranging from 7 to 12 Å, and 4th- or 5th-order interpolation. Each point in the four plots above represents the results for a particular set of Ewald parameters: the root mean squared error in electrostatic energy or instantaneous pressure (after removal of any bias) is plotted against the average force rmsd for all 20 snapshots. The division of the points into two groups stems from the different values of D_{tol}. Crosshairs in each plot intersect at the error in force and error in pressure or energy obtained using the default AMBER parameters ($\mu \leq 1$ Å, 4th order interpolation, $L_{cut} = 8.0$ Å, and $D_{tol} = 1.0 \times 10^{-5}$) for each system. Although the errors in pressure may appear large, only the amplitude is plotted and after removal of bias the error in pressure from StME calculations can be positive or negative. The root mean squared deviations in the instantaneous pressure obtained for streptavidin and COX-2 over the course of each simulation were both 90 bar, fluctuating about an average of 1 bar; in this sense, the errors in pressure are a miniscule amount of extra noise.

eters to obtain either AMBER or CHARMM accuracy on the four test cases.

In contrast to single-processor performance, parallel scaling is difficult to predict. We expect that the performance advantage of StME will carry over into simulations on small numbers of processors and that the ability to reduce the overall FFT workload without increasing the direct space workload will give StME another advantage in highly parallel applications. The optimal parameters will change with the number of processors, and as shown in Figure 11, StME offers many choices. Parallel implementations of the SPME algorithm have been extensively optimized for parallel scaling by multiple independent groups;[13–15] while StME can likely benefit from much of this progress, it will take some effort to devise a parallel StME implementation that is as finely tuned to make a fair comparison with the best SPME implementations.

6. Discussion

In this Article, we have reviewed and renewed an old technique, interlacing, for improving the accuracy of particle: mesh calculations. While the technique was originally used to reduce the memory requirements of such calculations at the expense of simulation speed, our new implementation Staggered Mesh Ewald seems to confer some benefits to overall speed on modern computers. Nowadays, computer memory is plentiful but the ability to use smaller meshes may help to reduce the total communication cost of simulations in parallel applications. We will now discuss how mesh staggering might benefit other Ewald mesh methods, Poisson solvers, and molecular simulations.

6.1. Staggered Meshes for Other Poisson Solvers. Previously, mesh staggering has been shown to be effective with the Particle:Particle Particle:Mesh (P³M) method,[17] and here, we have shown it to be effective with the Smooth Particle Mesh Ewald (SPME) method. The principal difference between these electrostatic methods is the shape of the charge smoothing function used in the mesh calculation: P³M uses a spherical hypercone, whereas SPME uses a spherical Gaussian, which confers some advantage in accuracy[9] because the Gaussians are better at conserving the total amount of charge on the mesh. We expect that mesh staggering will also improve the accuracy of Gaussian Split Ewald (GSE) calculations,[9] which are nearly identical to
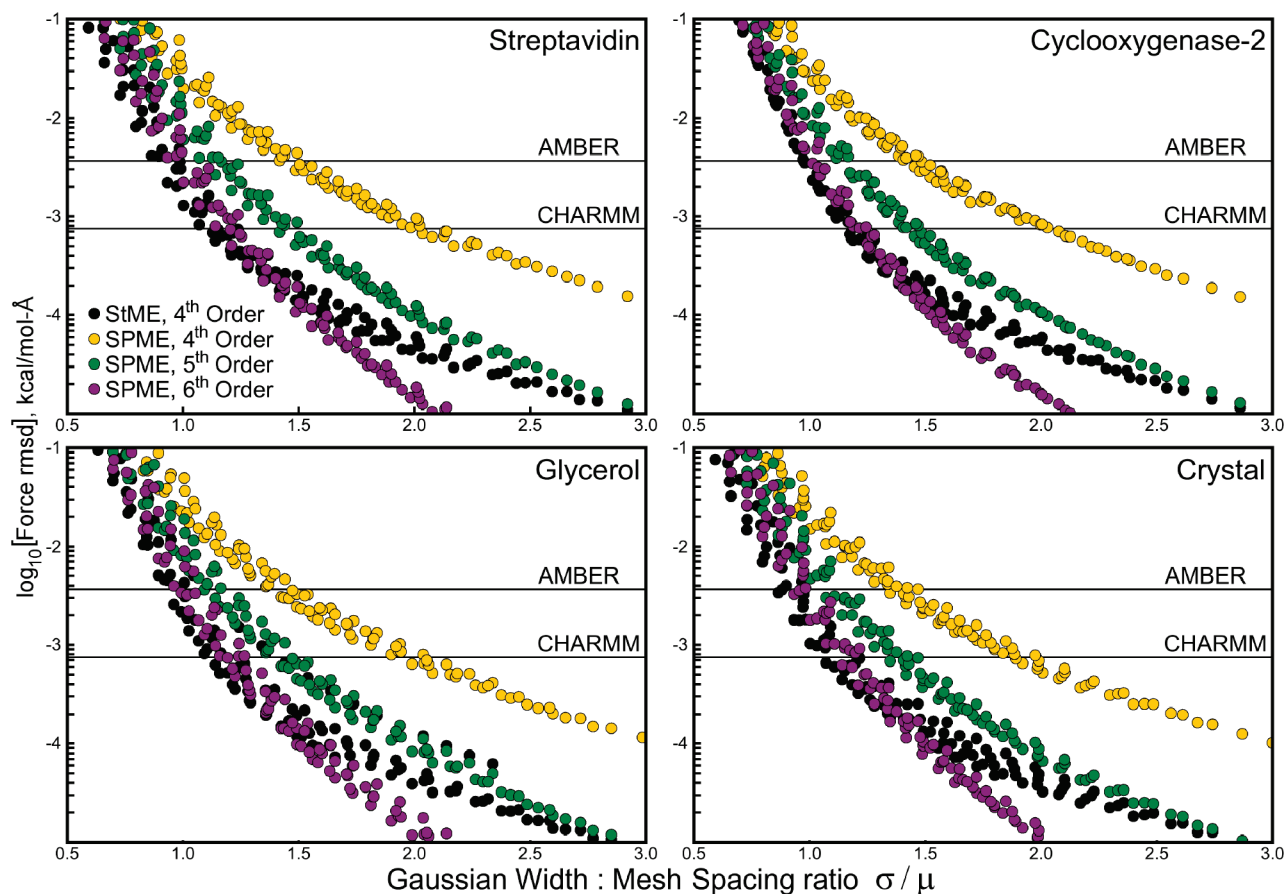
Figure 10. Accuracy of forces as a function of the ratio of charge smoothing to mesh spacing. In each of four systems, the accuracy of forces was computed for a range of values of direct space cutoff L_{cut} and mesh spacing μ. For all calculations, the direct sum tolerance D_{tol} was set to 1.0×10^{-9}, implying that errors in the forces came almost exclusively from the reciprocal space calculation. Equation S.7 of the Supporting Information was used to obtain the width of the Gaussian charge smoothing function in each calculation. Each plot is marked according to the "AMBER" and "CHARMM" accuracy standards, assuming that the reciprocal space calculation must create errors not in excess of half the level of each standard. The accuracy of the reciprocal space calculations as a function of σ/μ is a logical way to compare StME and SPME with different orders of interpolation: it indicates how aggressively the charges must be smoothed and hence provides an indication of how long L_{cut} must be for a particular μ. By this metric, StME with 4th order interpolation performs slightly better than SPME with 6th order interpolation to obtain the levels of accuracy sought in most molecular simulations.

SPME except in the function used to interpolate particles to the mesh (SPME uses a B-spline, whereas GSE uses another Gaussian, but B-splines in fact converge to Gaussians in the limit of high-interpolation order[24]).

Another possible application of mesh staggering is to real-space variants on the SPME method, which use the same particle interpolation and charge smoothing functions but solve Poisson's equation in real space.[10,11] These methods may prove more scalable than FFT-based methods on very high numbers of processors, but the principal drawback of these methods is the cost of smoothing the charge density in real space, a function of the number of mesh points. Staggered meshes can not only reduce the number of mesh points needed but also improve the efficiency of the charge smoothing procedure because the distances between corresponding mesh points are identical on both meshes. Such improvements may help close the gap between real-space and FFT-based Poisson solvers in Ewald calculations.

Fast Multipole methods (FMMs)[25,26] receive attention for the same reasons as real-space based Poisson solvers: the

promise of $O(N)$ scaling and also exponentially reduced communication requirements as the interactions become increasingly long-ranged. While FMMs continue to be slower than FFT-based Poisson solvers for condensed phase molecular systems, as with real-space Poisson solvers, considerable progress has been made in recent years. It is possible that staggering the hierarchy of meshes used by FMMs may confer the same benefits as staggering the one mesh used by SPME or P³M, again helping to close the performance gap between these methods and the standard particle:mesh techniques used in most molecular simulations.

While mesh staggering may have utility in other Poisson solvers, other approximations that have proven useful in standard Poisson solvers may be of utility in Staggered Mesh Ewald. In particular, the use of spherically truncated FFTs,[27] discarding very low-frequency modes in Fourier space much as interactions in the tail of the direct space sum are discarded in standard SPME, can decrease the cost and communication requirements of the FFT needed to take the charge Q mesh into Fourier space. By using spherically truncated FFTs and harmonic averaging[17] in the context of Staggered Mesh
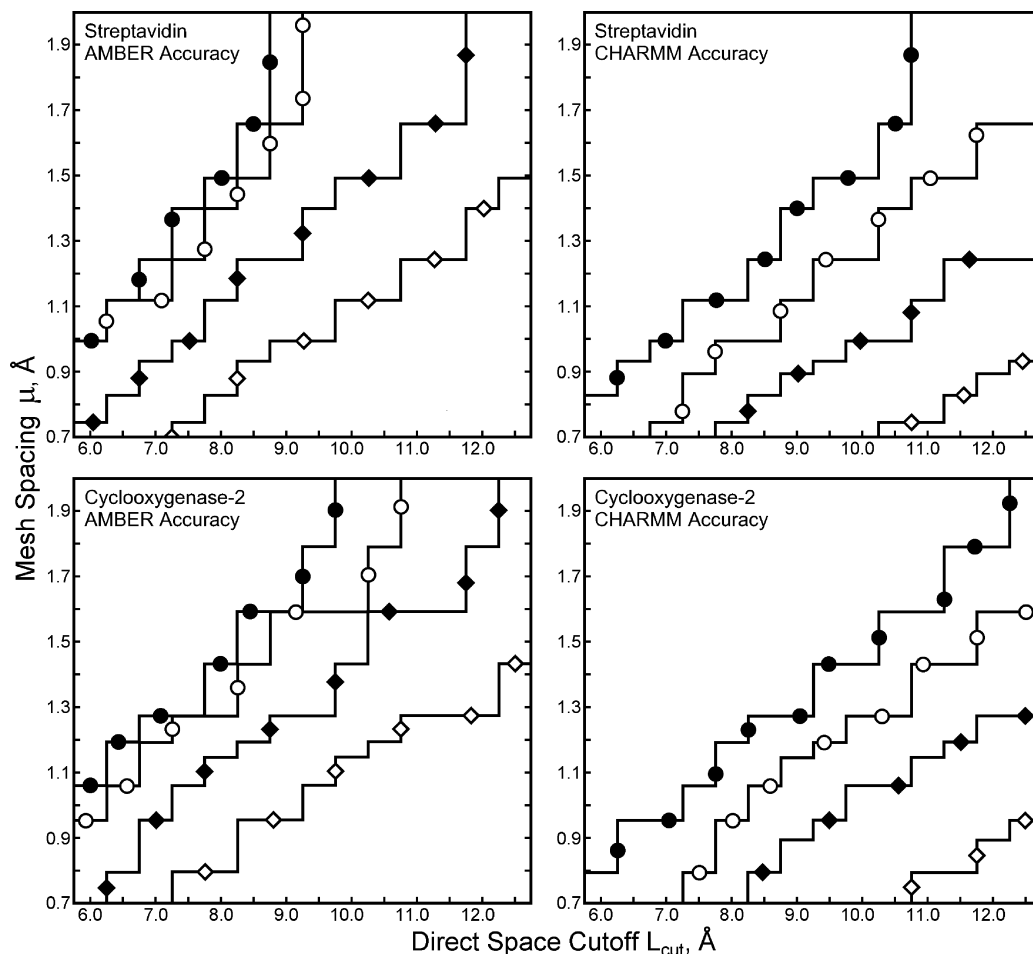
Figure 11. Ewald parameters yielding AMBER or CHARMM levels of accuracy in the streptavidin and COX-2 test cases. By scanning L_{cut} and μ for different orders of interpolation and optimizing D_{tol} for greatest accuracy in each case, we were able to determine the region of the L_{cut} and μ parameter space on which SPME or StME give acceptable levels of accuracy. The format of boundary lines in each panel follows Figure 6: diamonds denote the SPME method and circles denote the StME method, while open and filled symbols denote 4th and 5th order interpolation, respectively. Values of L_{cut} and μ below and to the right of each boundary line produce accurate forces according to the standard listed in each panel.

Table 2. Timings for Optimal Smooth Particle Mesh Ewald (SPME) Parameters and Estimated Timings for Staggered Mesh Ewald (StME) Parameters[a]

method	run parameters			timings			
	L_{cut}, Å	D_{tol}	mesh size	dir.[b]	rec.[c]	Total[d]	Factor[e]
			Streptavidin, AMBER accuracy				
SPME	8.5	4.0×10^{-6}	$96 \times 96 \times 96$	362	123	584	1.00
StME	6.5	4.0×10^{-6}	$80 \times 80 \times 80$	244	156	498	1.17
			Streptavidin, CHARMM accuracy				
SPME	10.0	5.0×10^{-7}	$120 \times 120 \times 120$	482	218	792	1.00
StME	8.5	8.0×10^{-7}	$80 \times 80 \times 80$	360	156	615	1.29

[a] Each system, described in Table 1, was run for 1000 steps in the NVT ensemble using a 1 fs time step, Berendsen thermostat, 10 Å cutoff on Lennard-Jones interactions, 2 Å nonbonded pairlist buffer, and the stated electrostatic parameters. The internal geometry of water molecules was constrained by SETTLE;[36] the lengths of other bonds to hydrogen were constrained by SHAKE.[37] Reciprocal space electrostatics were computed at every time step using the stated parameters and 4th-order interpolation. Timings for StME were estimated by doubling the reciprocal space calculation time of an SPME calculation run with the same parameters. This test used 4th-order interpolation exclusively because higher orders are very rarely used in practice and many codes, including the PMEMD version used for this test, use optimized routines for 4th-order interpolation. While we were able to obtain better overall run times by using higher orders of interpolation in the SPME runs, it would also be possible to improve the efficiency of StME runs if the two reciprocal space calculations were able to share data. This test is meant to offer a basic estimate of the efficiency of StME. [b] Direct space interaction computation time, including van der Waals interactions (all timings are in seconds). [c] Reciprocal space computation time. [d] Total simulation time, including nonbonded pairlist updates and bonded atom force calculations. [e] Overall rate of simulation, relative to SPME.

Ewald, the communication requirements and cost of evaluating the FFTs for the reciprocal space sum might be reduced even further.

6.2. Staggered Mesh Ewald for Highly Parallel Applications. We have not presented results for the performance of StME in the context of parallel molecular dynamics

Table 3. Timings for Optimal Smooth Particle Mesh Ewald (SPME) Parameters and Estimated Timings for Staggered Mesh Ewald (StME) Parameters[a]

method	run parameters			timings			
	L_{cut}, Å	D_{tol}	mesh size	dir.[b]	rec.[c]	total[d]	factor[e]
glycerol, AMBER accuracy							
SPME	8.5	5.0×10^{-6}	$72 \times 72 \times 90$	219	86	369	1.00
StME	8.0	7.0×10^{-6}	$48 \times 48 \times 60$	200	68	332	1.11
glycerol, CHARMM accuracy							
SPME	10.0	5.0×10^{-7}	$96 \times 96 \times 120$	284	190	534	1.00
StME	8.5	9.0×10^{-7}	$64 \times 64 \times 80$	222	119	405	1.32
crystal, AMBER accuracy							
SPME	8.0	7.0×10^{-6}	$90 \times 96 \times 96$	487	117	747	1.00
StME	7.5	8.0×10^{-6}	$64 \times 60 \times 64$	462	100	700	1.07
crystal, CHARMM accuracy							
SPME	10.0	1.0×10^{-6}	$120 \times 108 \times 120$	588	200	926	1.00
StME	9.0	1.0×10^{-6}	$72 \times 64 \times 72$	540	117	795	1.16
cyclooxygenase-2, AMBER accuracy							
SPME	8.5	5.0×10^{-6}	$120 \times 120 \times 120$	614	330	1127	1.00
StME	7.0	5.0×10^{-6}	$100 \times 100 \times 100$	458	417	1056	1.07
cyclooxygenase-2, CHARMM accuracy							
SPME	10.0	8.0×10^{-7}	$160 \times 160 \times 160$	782	719	1696	1.00
StME	9.0	8.0×10^{-7}	$100 \times 100 \times 100$	675	421	1281	1.32

[a] The format, molecular dynamics benchmark protocol, and labeling follows Table 2. StME performs better than SPME across all systems studied, though parallel scaling for an optimized StME implementation has not yet been tested. [b] Direct space interaction computation time, including van der Waals interactions (all timings are in seconds). [c] Reciprocal space computation time. [d] Total simulation time, including nonbonded pairlist updates and bonded atom force calculations. [e] Overall rate of simulation, relative to SPME.

simulations because we do not yet have a working implementation in this respect. We have shown that StME offers moderate performance improvements on single-processor simulations, which can be expected to carry over into parallel applications on small numbers of processors. The scaling of highly parallel algorithms is difficult to predict, but we will address three of the most critical aspects of the reciprocal space calculation with respect to parallel implementations and StME.

On large numbers of processors, the scaling of the Ewald reciprocal space calculation is limited primarily by the number of messages that must be passed between processors to accomplish the FFT operations for convoluting the charge mesh Q with the reciprocal space pair potential $\theta^{(rec)}$. If P nodes are working together to compute an FFT, each node must send its part of the FFT data to all other nodes, and receive FFT data from all other nodes. (It is for this reason that most codes devote a subset of processors to the reciprocal space calculation.) Because StME requires up to 40% less FFT work than regular SPME, the FFTs could be performed on a smaller subset of processors, implying fewer messages to pass.

But, reducing the amount of mesh data can imply other communication costs. A second important factor in the cost of a parallel reciprocal space calculation is the cost of communicating the coordinates and identities of atoms in order to construct the charge mesh Q, before any FFTs take place at all. In SPME with nth order interpolation, each atom influences a rectangular region of the mesh Q that is $n\mu$ points on each side. As n or μ increases, more atoms must be therefore imported from further away in order to consruct Q. As was shown in the results, StME with fourth order interpolation produces similar accuracy to SPME with sixth

order interpolation, all other parameters being equal. StME could therefore make use of a large μ such as 1.5 Å with smaller import regions for constructing Q than SPME would require. However, the import regions would still be somewhat larger than those required by an SPME calculation using the typical $\mu = 1$ Å.

A third factor that influences the cost of a parallel reciprocal space calculation is the actual cost of constructing Q and then interpolating forces from the potential $Q \star \theta^{(rec)}$. These particle ↔ mesh operations, which can be more expensive than the FFTs themselves (data not shown), are typically performed on the same processors that will do the convolution $Q \star \theta^{(rec)}$. Because StME essentially doubles the cost of the particle ↔ mesh operations, it may be difficult to reduce the number of processors devoted to the FFT operations. However, the particle ↔ mesh operations can be done on the more numerous processors devoted to the direct space calculation so that the grid data itself could be communicated to a subset of processors for computing the convolution. NAMD[15] is already equipped to run traditional SPME calculations by passing mesh data, not coordinates, to reciprocal space processors when the highest possible scaling is desired; such a reorganization may be necessary to make StME beneficial to highly parallel applications.

6.3. Continued Improvement of Ewald Mesh Methods. Ewald mesh methods will likely remain an important tool for molecular simulations well into the future. Briefly, calculating long-ranged Coulomb electrostatics currently accounts for a majority of the total simulation time, and will continue to do so even as classical models begin to incorporate other charge geometries and explicit polarization effects. While quantum effects are undoubtedly important

for the interaction of charges at very short-range, the Coulomb approximation quickly takes over even on molecular scales. Furthermore, numerous studies have shown that the periodic boundary conditions enforced by Ewald mesh methods are relatively benign, especially in comparison to some alternatives. We will describe the rationale for continued development of Ewald electrostatics in more detail, then outline other avenues to accelerating Ewald calculations which we hope will dramatically accelerate traditional Ewald mesh calculations and serve as a powerful complement to the Staggered Mesh Ewald method.

The majority of the computational effort in molecular dynamics simulations is devoted to electrostatic nonbonded interactions, whereas the calculation of van der Waals dispersion interactions, typically by Lennard-Jones potentials, is minor in comparison. Primarily, this is because electrostatic interactions are very long ranged. Moreover, because of the form of the electrostatic potential, analytic electrostatic force computations require not only a divide operation but also an expensive square root operation to obtain the quantity $[k_{coul}q_iq_j/r_{ij}^3]\mathbf{r}_{ij}$, where k_{coul} is Coulomb's constant, q_i and q_j represent charges of the atoms i and j, and $\mathbf{r}_{ij}$ is the vector between the charges. In contrast, the Lennard-Jones force requires only a divide operation to compute the quantity $[A_{ij}/r_{ij}^8 + B_{ij}/r_{ij}^{14}]\mathbf{r}_{ij}$, where A_{ij} and B_{ij} are constants. A third factor that makes electrostatic calculations dominate the cost of simulations is a peculiarity of current water models, most of which give Lennard-Jones attributes to only the oxygen atoms while placing charges on at least three sites. Because water makes up the majority of the system in most simulations of solvated proteins, there can be many more electrostatic interactions than Lennard-Jones interactions for a given cutoff.

While point charges may not be an adequate representation of atomic charge distributions at close range,[28] Ewald mesh methods are also compatible with other charge geometries. One must merely recall that the direct space interactions are a modification to the electrostatic potential of the smoothed charge distribution computed in the reciprocal space calculation, as discussed in the Introduction and Methods. The direct space modification can just as easily be used to extract interactions of distributed charges, so long as the interaction of two charges in the actual system and the interaction of two Gaussian charges converge at the direct space cutoff. Even if it does not perfectly describe the interaction of subatomic particles at close range, Coulomb's law is still valid for the interaction of charges on the nanometer scale. Therefore, long-ranged electrostatic methods such as the Ewald sum will continue to be essential for molecular simulations, even as new force fields with different local electrostatic approximations and even explicit polarization effects[29] come into use.

There is debate over whether periodicity imposed by Ewald electrostatics is suitable for molecular simulationss,[30,31] and while such a representation may be much more appropriate for crystal lattice simulations,[32] periodic boundary conditions are a very practical solution for simulations of proteins in boxes of water as well. While there are relevant concerns when using periodic boundary conditions with very

small systems,[33] finite size effects are by no means limited to periodic systems. Simulations performed in both periodic and nonperiodic unit cells such as droplets[34] or ice shells[35] suggest that periodic boundary conditions are as good or better than numerous alternatives.

Given the importance of Ewald mesh methods to molecular simulations, further developments that permit the use of coarser meshes or reduce the required number of direct space computations are of great interest. In this communication, we have analyzed the errors arising from a coarse mesh in terms of self-image forces and pair interaction force errors. The self-image forces we identified can be corrected on a per-atom basis for low-density plasma simulations, but the pair interaction force errors arising from a coarse mesh require more extensive corrections in condensed-phase simulations. Our solution was to introduce a second mesh calculation, staggered relative to the original. It may also be possible to modify the form of the Ewald "switching" function used to make the transition between the reciprocal space electrostatic potential and the direct space modification. The form used in all Ewald mesh methods to date, most apparent in Equation 1, is dictated by the form of the charge smoothing function, a Gaussian as described in Equation S1 of the Supporting Information. The typical direct space potential satisfies the most important property of an Ewald switching function in that it smoothly vanishes within a reasonable distance, while the associated Gaussian function ensures that charges can be mapped to a mesh with reasonable accuracy. However, it may be possible to design new charge smoothing and potential switching functions that map charges more accurately to coarser meshes or vanish more rapidly. We are pursuing new ways to satisfy these criteria and expect the results to be generally useful for all types of Ewald mesh calculations.

Supporting Information Available: Descriptions of the relationships between Gaussian charge smoothing width σ, direct sum tolerance D_{tol}, and the direct space cutoff L_{cut}, detailed description of the Ewald reciprocal space calculation, investigation of the sources of self-image force errors, pair interaction force errors, and biased energy and virial estimates inherent in SPME reciprocal space calculations using coarse meshes. This material is available free of charge via the Internet at http://pubs.acs.org.

References

(1) Beck, D. A. C.; Daggett, V. Methods for Molecular Dynamics Simulations of Protein Folding/Unfolding in Solution. *Methods* **2004**, *34*, 112–120.

(2) 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.

Staggered Mesh Ewald: Two Grids Are Better than One

J. Chem. Theory Comput., Vol. 5, No. 9, 2009 **2337**

(3) DeLeeuw, S. W.; Perram, J. W.; Smith, E. R. Simulation of Electrostatic Systems in Periodic Boundary Conditions. I. Lattice Sums and Dielectric Constants. *Proc. R. Soc. Lond. Ser. A* **1980**, *373*, 27–56.

(4) Yonetani, Y. a Severe Artifact in Simulation of Liquid Water Using a Long Cut-off Length: Appearance of a Strange Layer Structure. *Chem. Phys. Lett.* **2005**, *406*, 49–53.

(5) Patra, M.; Karttunen, M.; Hyvönen, M. T.; Falck, E.; Lindqvist, P.; Vattulainen, I. Molecular Dynamics Simulations of Lipid Bilayers: Major Artifacts Due to Truncating Electrostatic Interactions. *Biophys. J.* **2003**, *84*, 3636–3645.

(6) Pollock, E. L.; Glosli, J. Comments on P³M, FMM, and the Ewald Method for Large Periodic Coulombic Systems. *Comput. Phys. Commun.* **1996**, *95*, 93–110.

(7) Hockney, R. W.; and Eastwood, J. Collisionless Particle Models. In *Computer Simulation Using Particles*; Taylor and Francis Group: New York, 1988; pp. 260−291.

(8) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. H. A Smooth Particle Mesh Ewald Method. *J. Chem. Phys.* **1995**, *103*, 8577–8593.

(9) Shan, Y.; Klepeis, J. L.; Eastwood, M. P.; Dror, R. O.; Shaw, D. E. Gaussian Split Ewald: A Fast Ewald Mesh Method for Molecular Simulation. *J. Chem. Phys.* **2005**, *122*, 054101.

(10) Sagui, C.; Darden, T. Multigrid Methods for Classical Molecular Dynamics Simulations of Biomolecules. *J. Chem. Phys.* **2001**, *114*, 6578–6591.

(11) Beckers, J. V. L.; Lowe, C. P.; De Leeuw, S. W. An Iterative PPPM Method for Simulating Coulombic Systems on Distributed Memory Parallel Computers. *Mol. Simulat.* **1998**, *20*, 369–383.

(12) Darden, T.; York, D.; Pedersen, L. Particle Mesh Ewald: An N · log(N) Method for Ewald Sums in Large Systems. *J. Chem. Phys.* **1993**, *98*, 10089–10092.

(13) Case, D. A.; Darden, T. A.; Cheatham, T. E., III; Simmerling, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Crowley, M.; Walker, R. C.; Zhang, W.; Merz, K. M.; Wang, B.; Hayik, S.; Roitberg, A.; Seabra, G.; Kolossváry, I.; Wong, K. F.; Paesani, F.; Vanicek, J.; Wu, X.; Brozell, S. R.; Steinbrecher, T.; Gohlke, H.; Yang, L.; Tan, C.; Mongan, J.; Hornak, V.; Cui, G.; Mathews, D. H.; Seetin, M. G.; Sagui, C.; Babin, V.; Kollman, P. A. *AMBER 10*; University of California, San Francisco: San Francisco, CA, 2008.

(14) Bowers, K. J.; Chow, E.; Xu, H.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossváry, I.; Moraes, M. A.; Sacerdoti, F. D.; Salmon, J. K.; Shan, Y.; and Shaw, D. E. Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters. In *Conference on High Performance Networking and Computing*; Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, Tampa, FL, USA 2006; Association for Computing Machinery: New York 2006.

(15) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L.; Sand Chulten, K. Scalable Molecular Dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781–1802.

(16) Chen, L.; Langdon, B.; Birdsall, C. K. Reduction of the Grid Effects in Simulation Plasmas. *J. Comput. Phys.* **1974**, *14*, 200–222.

(17) Eastwood, J. Optimal P³M Algorithms for Molecular Dynamics Simulations. In *Computational Methods in Classical and Quantum Physics*; Hooper, M. B., Ed.; Advance Publications Ltd: London, U.K., 1976; pp 206−228.

(18) Hyre, D. E.; Le Trong, I.; Merritt, E. A.; Green, N. M.; Stenkamp, R. E.; Stayton, P. S. Wildtype Core-Streptavidin with Biotin at 1.4 Å. *Protein Sci.* **2006**, *15*, 459–467.

(19) Smith, G. D.; Blessing, R. H.; Ealick, S. E.; Fontecilla-Camps, J. C.; Hauptman, H. A.; Housset, D.; Langs, D. A.; Miller, R. Ab Initio Structure Determination and Refinement of a Scorpion Protein Toxin. *Acta Crystallogr. D* **1997**, *53*, 551–557.

(20) Kiefer, J. R.; Pawlitz, J. L.; Moreland, K. T.; Stegeman, R. A.; Gierse, J. K.; Stevens, A. M.; Goodwin, D. C.; Rowlinson, S. W.; Marnett, L. J.; Stallings, W. C.; Kurumbail, R. G. Structural Insights into the Stereochemistry of the Cyclooxygenase Reaction. *Nature* **2000**, *405*, 97–101.

(21) Chelli, R.; Procacci, P.; Cardini, G.; Cella; Valle, R. G.; Califano, S. Glycerol Condensed Phases Part I. A Molecular Dynamics Study. *Phys. Chem. Chem. Phys.* **1999**, *1*, 871–877.

(22) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of Multiple Amber Force Fields and Development of Improved Protein Backbone Parameters. *Proteins* **2006**, *65*, 712–725.

(23) Crocker, M. S.; Hampton, S. S.; Matthey, T.; Izaguirre, J. A. MDSIMAID: Automatic Parameter Optimization in Fast Electrostatic Algorithms. *J. Comput. Chem.* **2005**, *26*, 1021–1031.

(24) Unser, M.; Aldourbi, A.; Eden, M. On the Asymptotic Convergence of B-Spline Wavelets to Gabor Functions. *IEEE Trans. Inf. Theory* **1992**, *38*, 864–872.

(25) Lu, B.; Cheng, X.; McCammon, J. A. "New-Version-Fast-Multipole-Method" Accelerated Electrostatic Calculations in Biomolecular Systems. *J. Comput. Phys.* **2007**, *226*, 1348–1366.

(26) Lu, B.; Cheng, X.; Huang, J.; McCammon, J. A. Order *N* Algorithm for Computation of Electrostatic Interactions in Biomolecular Systems. *Proc. Natl. Acad. Sci. U. S. A.* **2006**, *103*, 19314–19319.

(27) Fang, B.; Martyna, G.; Deng, Y. A Fine Grained Parallel Smooth Particle Mesh Ewald Algorithm for Biophysical Simulation Studies: Application to the 6D-Torus QCDOC Supercomputer. *Comput. Phys. Commun.* **2007**, *177*, 362–377.

(28) Paricaud, P.; Predota, M.; Chialvo, A. A.; Cummings, P. T. From Dimer to Condensed Phases at Extreme Conditions: Accurate Predictions of the Properties of Water by a Gaussian Charge Polarizable Model. *J. Chem. Phys.* **2005**, *122*, 244511.

(29) Warshel, A.; Kato, M.; Pisliakov, A. V. Polarizable Force Fields: History, Test Cases, and Prospects. *J. Chem. Theory Comput.* **2007**, *3*, 2034–2045.

(30) Villareal, M. A.; Montich, G. G. On the Ewald Artifacts in Computer Simulations. The Test-Case of the Octaalanine Peptide with Charged Termini. *J. Biomol. Struct. Dyn.* **2005**, *23*, 135–142.

(31) Hünenberger, P. H.; McCammon, J. A. Ewald Artifacts in Computer Simulations of Ionic Solvation and Ion−Ion Interaction: A Continuum Electrostatics Study. *J. Chem. Phys.* **1999**, *110*, 1856–1872.

(32) Cerutti, D. S.; Le Trong, I.; Stenkamp, R. E.; Lybrand, T. P. Simulations of a Protein Crystal: Explicit Treatment of Crystallization Conditions Links Theory and Experiment in the Streptavidin-Biotin Complex. *Biochemistry* **2008**, *47*, 12065–12077.

(33) Weerasinghe, S.; Smith, P. E. A Kirkwood−Buff Derived Force Field for Mixtures of Urea and Water. *J. Phys. Chem. B* **2003**, *107*, 3891–3898.

(34) Freitag, S.; Chu, V.; Penzotti, J. E.; Klumb, L. A.; To, R.; Hyre, D.; Le Trong, I.; Lybrand, T. P.; Stenkamp, R. E.; Stayton, P. S. A Structural Snapshot of an Intermediate on the Streptavidin−Biotin Dissociation Pathway. *Proc. Natl. Acad. Sci. U. S. A.* **1999**, *96*, 8384–8389.

(35) Riihimäki, E. S.; Martínez, J. M.; Kloo, L. An Evaluation of Non-periodic Boundary Condition Models in Molecular Dynamics Simulations Using Prion Octapeptides As Probes. *THEOCHEM* **2005**, *760*, 91–98.

(36) Miyamoto, S.; Kollman, P. A. SETTLE: An Analytical Version of the SHAKE and RATTLE Algorithm for Rigid Water Models. *J. Comput. Chem.* **1992**, *13*, 952–962.

(37) Ryckaert, J. P.; Ciccotti, G.; Berendsen, H. J. C.; Hirasawa, K. Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of *n*-Alkanes. *J. Comput. Phys.* **1997**, *23*, 327–341.