JCTC Journal of Chemical Theory and Computation

Multi-Level Ewald: A Hybrid Multigrid/Fast Fourier Transform Approach to the Electrostatic Particle-Mesh Problem

David S. Cerutti* and David A. Case

Department of Chemistry and Chemical Biology, and BioMaPS Institute, Rutgers University, 610 Taylor Road, Piscataway, New Jersey 08854-8066

Received October 3, 2009

Abstract: We present a new method for decomposing the one convolution required by standard Particle−Particle Particle-Mesh (P^3M) electrostatic methods into a series of convolutions over slab-shaped subregions of the original simulation cell. Most of the convolutions derive data from separate regions of the cell and can thus be computed independently via FFTs, in some cases with a small amount of zero padding so that the results of these subproblems may be reunited with minimal error. A single convolution over the entire cell is also performed, but using a much coarser mesh than the original problem would have required. This “Multi-Level Ewald” (MLE) method therefore requires moderately more FFT work plus the tasks of interpolating between different sizes of mesh and accumulating the results from neighboring subproblems, but we show that the added expense can be less than 10% of the total simulation cost. We implement MLE as an approximation to the Smooth Particle-Mesh Ewald (SPME) style of P^3M and identify a number of tunable parameters in MLE. With reasonable settings pertaining to the degree of overlap between the various subproblems and the accuracy of interpolation between meshes, the errors obtained by MLE can be smaller than those obtained in molecular simulations with typical SPME settings. We compare simulations of a box of water molecules performed with MLE and SPME and show that the energy conservation, structural, and dynamical properties of the system are more affected by the accuracy of the SPME calculation itself than by the additional MLE approximation. We anticipate that the MLE method’s ability to break a single convolution into many independent subproblems will be useful for extending the parallel scaling of molecular simulations.

1. Introduction

Observing biochemical processes through computer simulations requires thorough equilibrium sampling of a protein or nucleic acid system with thousands of degrees of freedom. The quality of the molecular model is of utmost importance, but validation requires extensive simulations to yield precise results for properties such as equilibrium conformations,[1] crystallographic temperature factors,[2] binding energies,[3] and molecular folding rates.[4] The capabilities of molecular simulations and the models themselves therefore evolve in step with computer performance and parallel algorithm design.

The central challenge with parallel molecular dynamics algorithms is the treatment of electrostatic interactions. Because the electrostatic potential decays as the inverse distance, charged particles influence one another at long-range, implying a great deal of information sharing and potentially a great deal of algorithmic complexity, as much as $O(N^2)$ in the number of particles N. Particle $\leftrightharpoons$ mesh implementations of the Ewald sum,[5,6] and more generally Particle−Particle Particle-Mesh (P^3M) methods,[7] are popular choices for treating long-ranged electrostatic forces in

* Corresponding author phone: (732) 445-0334; fax: (732) 445-5958; e-mail: dcerutti@rci.rutgers.edu.

10.1021/ct900522g © 2010 American Chemical Society
Published on Web 12/29/2009

molecular simulations because of the favorable complexity of the algorithms, $O(M \log M)$ or $O(M)$, for a number of mesh grid points M depending on the choice of Poisson solver. For commodity hardware, Poisson solvers based on the fast-Fourier transforms (FFTs) are commonly used because of their computational efficiency,[5,8] which is so great that the parallel scaling of these approaches is still limited, on most clusters, to a few hundred processors. Most molecular dynamics codes meet high scaling targets by dedicating a subset of the processors to the FFT, but the number of messages and the amount of data that must be shared can still limit the total number of processors that can be devoted to the calculation and thus the maximum speed of molecular simulations.

Some recent variations of P^{3}M[9,10] make use of real-space Poisson solvers based on finite difference or multigrid methods. These approaches offer better algorithmic complexity ($O(M)$ for the real-space methods, versus $O(M \log M)$ for the FFT-based methods) and asymptotically better interprocessor communication in parallel calculations. However, because the real-space solvers require significantly more work to map the particles' charge density to the mesh and extract forces from the mesh, their only successful application has been on specialized hardware.[9]

Other strategies for solving particle $\leftrightharpoons$ mesh problems can be found in the broad class of Multilevel Summation methods pioneered by Brandt[11] and developed for molecular simulations by Skeel[12] and others, the Fast-Multipole Method,[13] and Adaptive P^{3}M techniques used in astrophysical gravity calculations.[14−16] These mesh refinement techniques, along with the basic P^{3}M method, may all be viewed as variations on the theme of smoothly and (in essence) isotropically splitting a long-ranged potential into short- and long-ranged components that can be accurately represented on meshes of different resolutions. Like multigrid Poisson solvers, when applied to condensed-phase systems, the mesh refinement methods also exchange computational effort for scaling benefits, but the simplicity of the algorithms and communication patterns makes these methods highly adaptable for applications on commodity hardware and general-purpose graphics processor units.

In this Article, we present an alternative method for replacing the one convolution required by traditional electrostatic P^{3}M solvers with a series of convolutions, each pertaining to a slab subdomain of the simulation cell, building up to a single convolution involving a coarsened mesh which describes the entire simulation cell. In contrast to the smooth splitting employed by other mesh refinement methods, our approach is to split the mesh-based potential sharply and anisotropically such that the individual components all contain discontinuities but can nonetheless recover a smooth potential when summed. This "Multi-Level Ewald" (MLE) approach produces the electrostatic potential in a single pass over all levels of the mesh; the extra computational effort is small.

We explore numerous strategies for manipulating the parameters of MLE scheme itself and the details of the associated particle $\leftrightharpoons$ mesh operations to tune the accuracy of the resulting forces and energies with small amounts of overlap between adjacent slabs. Approximating the reciprocal space convolution using MLE can incur scarcely more error in the resulting particle forces than would be obtained with an equivalent P^{3}M calculation. We expect that MLE can reduce the data communication requirements of molecular dynamics simulations for modern networked computing architectures and will prove adaptable for balancing communication loads when the network connectivity is heterogeneous.

2. Theory

2.1. The Problem of Computing Long-Ranged Electrostatics, the Ewald Solution, and Its Evolution into P^{3}M. The Ewald method can be summarized as splitting the calculation of the electrostatic energy of a periodic system of point (or otherwise highly localized) charges $E^{(\mathrm{coul})}$ into a "reciprocal space" sum describing the energy of a system of spherical Gaussian charges, which has identical coordinates to the system of interest, and a "direct space" sum, which modifies the energy of the reciprocal space sum to recover the energy of the system of point charges:

$$\begin{aligned} E^{(\mathrm{coul})} &= \frac{1}{2}\sum_{n}\sum_{i}\sum_{j\neq i}\frac{q_i q_j}{4\pi\varepsilon_0|\mathbf{n}\cdot\mathbf{L} + \mathbf{r}_{ij}|} \\ E^{(\mathrm{dir})} &\simeq \frac{1}{2}\sum_{i}\sum_{j\neq i}\frac{q_i q_j(1 - \mathrm{erf}(\beta|\mathbf{r}|))}{4\pi\varepsilon_0|\mathbf{r}_{ij}|} = \sum_{i}\sum_{j\neq i} q_i q_j \theta^{(\mathrm{dir})} \\ E^{(\mathrm{rec})} &= \frac{1}{2}\sum_{n}\sum_{i}\sum_{j\neq i}\frac{q_i q_j \mathrm{erf}(\beta|\mathbf{r}|)}{4\pi\varepsilon_0|\mathbf{n}\cdot\mathbf{L} + \mathbf{r}_{ij}|} = \sum_{i}\sum_{j\neq i} q_i q_j \theta^{(\mathrm{rec})} \\ E^{(\mathrm{coul})} &= E^{(\mathrm{dir})} + E^{(\mathrm{rec})} \end{aligned} \tag{1}$$

In these equations, $\mathbf{n}\cdot\mathbf{L}$ represents images of the unit cell throughout all space, i and j run over all charged particles in the system, $\mathbf{r}_{ij}$ is the distance between particles i and j, ε_0 is the permittivity of free space, and β is the "Ewald coefficient". The reciprocal and direct space sums, $E^{(\mathrm{rec})}$ and $E^{(\mathrm{dir})}$, obtain their names because each converges absolutely in Fourier (reciprocal) or real (direct) space, respectively. The splitting is necessary because a straightforward summation over many periodic images of all charges in the system will not converge absolutely.

In its original formulation, the Ewald method relies only on the positions of particles. The direct space calculation involves a loop over all particles with a nested loop over each particle's neighbors within a cutoff distance L_{cut} sufficient to give a convergent direct space sum. The reciprocal space calculation involves a loop over all particles with a nested loop that again involves all particles.

Splitting a potential into short- and long-ranged components is also the basis of Particle−Particle Particle-Mesh (P^{3}M) methods.[7] These similarities led Darden and colleagues to propose "Particle-Mesh Ewald"[6] as a special case of P^{3}M, which incorporated a Gaussian function for splitting the potential function. Many variants of this particular case have since been developed,[5,9,10] along with distinct approaches for optimizing the influence function that modulates the interaction of charges on the mesh.[17] In all of these methods, the basic procedure may be summarized: (1) assign charges to the density mesh Q using the positions and charges of particles plus a suitable particle → mesh interpolation kernel, (2) solve the field on the mesh by convolution with the mesh-based representation of the interparticle potential function, and (3) interpolate forces on all particles given the particle positions, charges, field values, and particle → mesh interpolation kernel.

The most common motivation for using particle-mesh strategies is to exploit the convolution theorem, which states that for two sequences of numbers f_1 and f_2,

$$f_1 \bigstar f_2 = \mathscr{F}^{-1}[\mathscr{F}(f_1) \cdot \mathscr{F}(f_2)] \tag{2}$$

Above, ★ represents a convolution, $\mathscr{F}(f)$ is the (fast) discrete Fourier transform (FFT) of f, $\mathscr{F}^{-1}(f)$ is the inverse FFT of f such that $\mathscr{F}^{-1}(\mathscr{F}(f)) = f$, and $\mathscr{F}(f_1) \cdot \mathscr{F}(f_2)$ is the element-wise product of $\mathscr{F}(f_1)$ and $\mathscr{F}(f_2)$.

The most popular variant of P^{3}M for electrostatics, Smooth Particle-Mesh Ewald (SPME),[5] makes use of cardinal B-splines[18] to map the system's charges to the mesh Q. An elegant derivation of the Fourier transform of the reciprocal space pair potential, $\mathscr{F}(\theta^{(\mathrm{rec})})$, is obtained by folding together Euler exponential splines in the mesh B (the Fourier-space representation of B-splines are Euler splines) and the Fourier-space representation of the Gaussian charge smoothing function W (the Fourier transform of a Gaussian is another Gaussian):

$$B(x, y, z) = |b(x)| \cdot |b(y)| \cdot |b(z)|$$

$$b(\eta_\alpha) = \frac{\exp(2\pi i(n-1)\eta_\alpha / g_\alpha)}{[\sum_{p=0}^{n-2} M_n(p+1)\exp(2\pi i p \eta_\alpha / g_\alpha)]} \tag{3}$$

$$W(x, y, z) = \frac{\exp(-4\pi^2 |\mathbf{k}|^2 \sigma^2)}{\pi V |\mathbf{k}|^2} \tag{4}$$

$$\mathscr{F}(\theta^{(\mathrm{rec})}) = B \cdot W \tag{5}$$

In the equations defining B and W, M_n represents a cardinal B-spline of order n, i is the square root of -1, α is one of the mesh dimensions x, y, or z, η_α is a displacement in the mesh dimension α, and g_α is the size of the mesh in α. Also, in the equation defining W, V is the volume of the simulation cell, σ is the rms of the Gaussian charge smoothing function (note that $\sigma = 1/(2\beta)$), and $\mathbf{k}$ is the displacement from the origin in Fourier space. (Readers should consult the original SPME reference[5] for a detailed presentation of the derivation of this approximation to $\theta^{(\mathrm{rec})}$ and particle ⇋ mesh interpolation using B-splines. We have provided the most important definitions here because they will be important later as we develop our new method.) After $\mathscr{F}(\theta^{(\mathrm{rec})})$ has been prepared, the electrostatic potential $U^{(\mathrm{rec})}$ is computed with only two FFT operations:

$$U^{(\mathrm{rec})} = \mathscr{F}^{-1}[\mathscr{F}(Q) \cdot \mathscr{F}(\theta^{(\mathrm{rec})})] \tag{6}$$

The electrostatic potential energy of the system $E^{(\mathrm{rec})}$ may then be obtained by element-wise multiplication of the charge density Q with the electrostatic potential:

$$E^{(\mathrm{rec})} = Q \cdot U^{(\mathrm{rec})} \tag{7}$$

This operation would be performed in real space and would require that a copy of the original charge density be saved before computing $U^{(\mathrm{rec})}$. To avoid this extra memory requirement, FFT-based Poisson solvers use an identity to obtain $E^{(\mathrm{rec})}$ during the element-wise multiplication in Fourier space, when the system virial is available as well. However, we emphasize the real-space expression for the energy as this will be necessary as we develop a replacement to the convolution step of P^{3}M methods in electrostatics.

2.2. Accurate Decomposition of the Mesh-Based Sum: The MLE Method. In the interest of improving the parallel scaling of P^{3}M methods for molecular electrostatics, we focused on improving the method in which $Q \bigstar \theta^{(\mathrm{rec})}$ is computed while preserving other aspects of the algorithm. Our approach was to split $\theta^{(\mathrm{rec})}$ into fine and coarse resolution components as shown in Figure 1. Rather than splitting the potential isotropically in terms of the absolute distance between points, however, the splitting is done anisotropically along planes perpendicular to one dimension, which we will call $\hat{x}$. The fine resolution potential $\theta_{\mathrm{h}}^{(\mathrm{rec})}$ exactly describes the interactions between any two mesh points separated by up to (and including) T_{cut} in $\hat{x}$, regardless of the distance between the points in the other unit cell dimensions $\hat{y}$ and $\hat{z}$. Conversely, the low resolution pair potential $\theta_{\mathrm{c}}^{(\mathrm{rec})}$ approximately describes the interactions of points separated by more than T_{cut} in $\hat{x}$, no matter their locations in $\hat{y}$ and $\hat{z}$. For convenience, we will refer to the set of mesh points that share the same coordinate in the $\hat{x}$ direction as a "page" of the mesh. The meshes of different resolutions used in our approximation will be referred to as different "levels" of mesh. Our convention is to call the finest resolution mesh the lowest level; the extent of the reciprocal space pair potential grows as the mesh level becomes higher.

With $\theta^{(\mathrm{rec})}$ split into two components, the convolution can be restated:

$$Q \bigstar \theta^{(\mathrm{rec})} \simeq Q \bigstar \theta_{\mathrm{h}}^{(\mathrm{rec})} + Q_{\mathrm{c}} \bigstar \theta_{\mathrm{c}}^{(\mathrm{rec})} = U_{\mathrm{h}}^{(\mathrm{rec})} + U_{\mathrm{c}}^{(\mathrm{rec})} \simeq U^{(\mathrm{rec})} \tag{8}$$

where Q_{c} is a coarsened charge mesh interpolated from Q. (Note that, while $\theta_{\mathrm{h}}^{(\mathrm{rec})}$ is sparse and $\theta_{\mathrm{c}}^{(\mathrm{rec})}$ contains a void where $\theta_{\mathrm{h}}^{(\mathrm{rec})}$ is nonzero, Q and Q_{c} are full: every point in Q_{c} is interpolated from the appropriate points in Q.) This approximation does not immediately reduce the communication requirements of computing $Q \bigstar \theta^{(\mathrm{rec})}$, but because $\theta_{\mathrm{h}}^{(\mathrm{rec})}$ is zero in all but a narrow region $2(T_{\mathrm{cut}}) + 1$ pages thick, the convolution $Q \bigstar \theta_{\mathrm{h}}^{(\mathrm{rec})}$ can be accomplished as a series of convolutions:

$$Q \bigstar \theta_{\mathrm{h}}^{(\mathrm{rec})} = \sum_{i=1}^{P} [Q_i \bigstar \theta_{\mathrm{h}}^{(\mathrm{rec})}] \tag{9}$$

where each submesh Q_i spans the simulation box in $\hat{y}$ and $\hat{z}$ and is zero-padded $2T_{\mathrm{cut}}$ pages in $\hat{x}$. We will refer to these submeshes as "slabs". Each of the P slabs of the lowest level mesh is therefore padded by $2T_{\mathrm{cut}}$ pages of zeroes, and each of the series of convolutions described in eq 9 can be computed independently. The padding is done so that FFTs

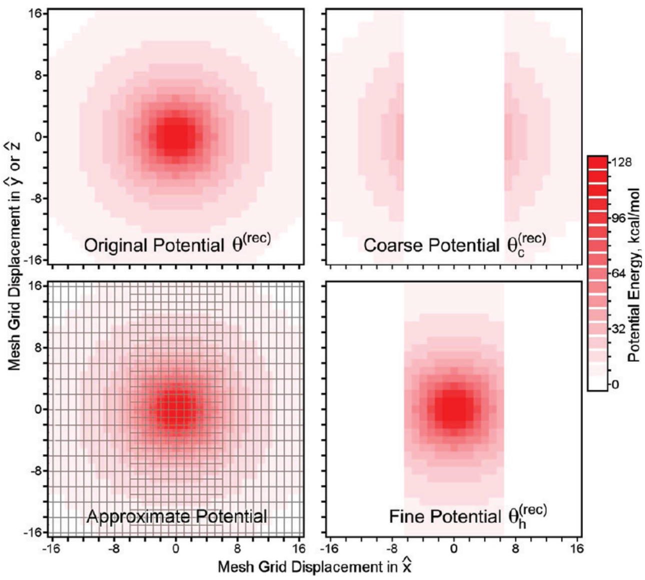

Figure 1. A multilevel mesh-based approximation to the Ewald reciprocal space pair potential. In Ewald mesh calculations, the reciprocal space pair potential $\theta^{(rec)}$ can be visualized in real space. At short-range, $\theta^{(rec)} \sim \mathrm{erf}(\beta|\mathbf{r}|)/(4\pi\varepsilon_0|\mathbf{r}|)$, where ε_0 is the permittivity of free space and erf is the error function. $\theta^{(rec)}$ was computed for a 96 Å cubic box on a mesh of 96^3 points; slices of the potential through the *xy* (or *xz*) planes are shown with varying intensities of red to indicate the magnitude. The color scale is deliberately coarse to make the potential isocontours apparent. $\theta^{(rec)}$ varies most rapidly along paths passing directly through the source at (0,0,0); paths that move tangentially to the source encounter much slower variations in $\theta^{(rec)}$. It is more feasible to approximate $\theta^{(rec)}$ with high- and low-resolution potentials $\theta_h^{(rec)}$ and $\theta_c^{(rec)}$ as shown, avoiding mesh ⇌ mesh interpolation along vectors pointed at the source as much as possible. $\theta_c^{(rec)}$ uses double the mesh spacing along the $\hat{y}$ and $\hat{z}$ axes, as indicated by the mesh overlay in the lower left panel, but the same spacing as $\theta_h^{(rec)}$ in the $\hat{x}$ direction. $\theta_c^{(rec)}$ therefore presents a fine mesh spacing for interpolating gradients of the true potential $\theta^{(rec)}$ when the true potential varies rapidly, but presents a coarse spacing when the true potential varies slowly.

may be used for the convolution without having charges near the *yz* faces of any slab "wrap around" and erroneously influence other parts of the same slab, and so that the influence of charges near the *yz* faces of each slab will be recorded as the results of all of these convolutions over slabs are then spliced back together to accumulate the approximation to $U^{(rec)}$. The electrostatic influence of charge density in slab Q_i on the neighboring slabs is recorded in its zero-padded pages. (Generally, the neighbors of Q_i are Q_{i-1} and Q_{i+1}, although Q_P and Q_1 are neighbors due to the periodicity of the unit cell.) The basic procedure is illustrated in Figure 2. In principle, convolutions with many radially symmetric potential functions could be split in this manner, although we focus on the case of the inverse distance kernel for application to biomolecular simulations. Numerous styles of P^{3}M are also compatible with this convolution splitting procedure; we have chosen to implement it within the Smooth Particle-Mesh Ewald style described above and call the new method "Multi-Level Ewald" (MLE).

In our MLE implementation, the coarsened reciprocal space pair potential $\theta_c^{(rec)}$ is obtained simply by extracting points from $\theta^{(rec)}$ at regular intervals of the coarsening factor, C_{yz}, in the $\hat{y}$ and $\hat{z}$ directions:

$$\theta_c^{(rec)}(i,j,k) = \begin{cases} \theta^{(rec)}(i, C_{yz}j, C_{yz}k) & ,i > T_{\mathrm{cut}} \text{ or } i < \, = g_x - T_{\mathrm{cut}} \\ 0 & ,\text{otherwise} \end{cases} \tag{10}$$

where *i*, *j*, and *k* represent coordinates in the $\hat{x}$, $\hat{y}$, and $\hat{z}$ directions, respectively, and g_x is the mesh size in the $\hat{x}$ direction. (Here, *k* should not be confused with **k** used earlier to describe a vector in Fourier space.) While this approach may appear to discard much of the information present in $\theta^{(rec)}$, we will show that it can produce very high accuracy depending on the other MLE parameters. We wrote an optimization procedure to try and improve the coarsened reciprocal space pair potential mesh by using steepest descent optimization to adjust the values of $\theta_c^{(rec)}$ at individual mesh points and minimize the root mean squared (rms) error in the approximate $U^{(rec)}$. This approach could only reduce the error rate of MLE calculations by about 2% (data not shown) and was not given further consideration in these studies.

Similar to the construction of $\theta_c^{(rec)}$, Q_c is interpolated from Q using cardinal B-splines[18] similar to those used for particle ⇌ mesh interpolation in standard SPME. However, mesh ⇌ mesh interpolation is a two-dimensional process as the mesh resolution is only reduced in $\hat{y}$ and $\hat{z}$ and maintained in $\hat{x}$. Each page of the mesh Q_c is interpolated from the corresponding page of Q. As we will show in the Results, it can be advantageous to use relatively high values of the order of mesh ⇌ mesh interpolation $I^{(mm)}$ as opposed to the order of particle ⇌ mesh interpolation $I^{(pm)}$. In the same way that higher values of $I^{(pm)}$ improve the accuracy of SPME calculations, higher values of $I^{(mm)}$ improve the accuracy of the MLE approximation; however, whereas the cost of an SPME calculation scales as the cube of $I^{(pm)}$ because each particle has a different alignment to the mesh, the regularity of the mesh ⇌ mesh interpolation makes it separable in each dimension, and thus the operation scales merely as $I^{(mm)}$.

While $\theta_h^{(rec)}$ will typically span a small region of the simulation box, if $\theta_c^{(rec)}$ were to span the rest it might be impractical to compute $Q_c \star \theta_c^{(rec)}$ as a series of convolutions. However, it is still possible to add more mesh levels by splitting $\theta_c^{(rec)}$ into its own "fine" and "coarse" resolution components $\theta_{c,1}^{(rec)}$, $\theta_{c,2}^{(rec)}$, ..., $\theta_{c,n}^{(rec)}$, in the same manner that the original $\theta^{(rec)}$ was split (see Figure 3). The most general expression of the MLE method is then:

$$Q \star \theta^{(rec)} = \sum_{i=1}^{P_1} [Q_i \star \theta_h^{(rec)}] + \sum_{j=2}^{L-1} \sum_{k=1}^{P_j} [Q_{c,j,k} \star \theta_{c,j}^{(rec)}] + Q_{c,L} \star \theta_{c,L}^{(rec)} \tag{11}$$

The scheme above involves $L - 1$ coarsened meshes with as many distinct coarsening factors. In general, a single convolution of the highest level charge mesh with the coarsest component of the reciprocal space pair potential must be performed, involving data collected over the entire simulation cell. However, because the mesh spacing in the

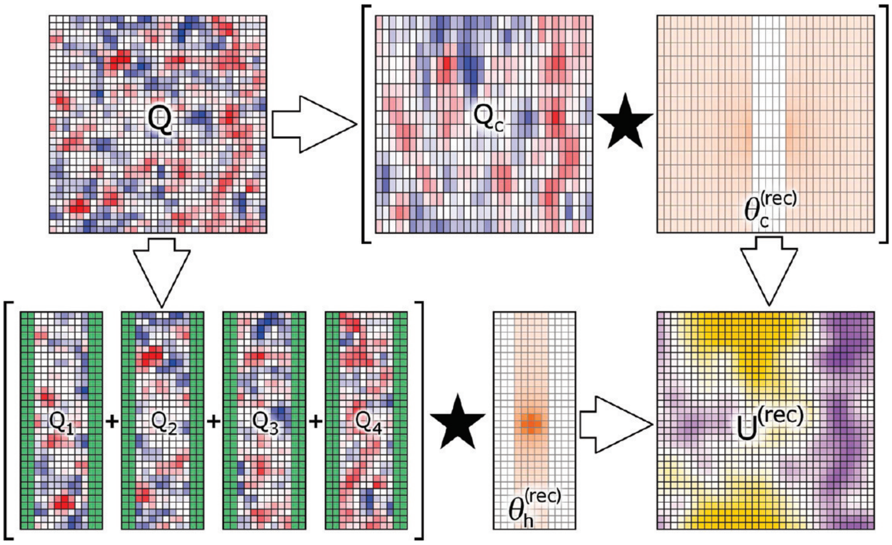

Figure 2. Illustration of the Multi-Level Ewald convolution procedure. In typical Smooth Particle-Mesh Ewald (SPME) calculations, the charge mesh Q is convoluted with $\theta^{(rec)}$ to arrive at the reciprocal space electrostatic potential $U^{(rec)}$. In Multi-Level Ewald (MLE), this single convolution is replaced with many smaller ones. In a basic two-level variation of MLE, the mesh Q is split, with no interpolation, into multiple subregions (slabs) $Q_1...Q_L$ as shown. The slabs are then zero-padded, as highlighted in green in the diagram, so that they may be convoluted with the high-resolution reciprocal space pair potential $\theta_h^{(rec)}$, which is itself extracted from $\theta^{(rec)}$ without interpolation. The coarsened charge mesh Q_c is interpolated from Q and convoluted with the coarsened reciprocal space pair potential $\theta_c^{(rec)}$ (see Figure 1). An electrostatic potential at the resolution of the fine mesh is then interpolated from the result of $Q_c★\theta_c^{(rec)}$ to complete the approximation of $Q★\theta^{(rec)}$. This figure was made using an actual MLE calculation on a 32 Å^3 box of 4000 randomly distributed ions. The color scales are not given as the diagram is qualitative, but in the meshes Q, Q_c, and $Q_1...Q_4$ red and blue signify negative and positive charge, the intensity of orange signifies the intensity of the pair potential, and purple and gold signify negative and positive electrostatic potential in the resulting $U_c^{(rec)}$. Each colored pixel corresponds to a point in a plane cutting through the mesh in the actual MLE calculation.

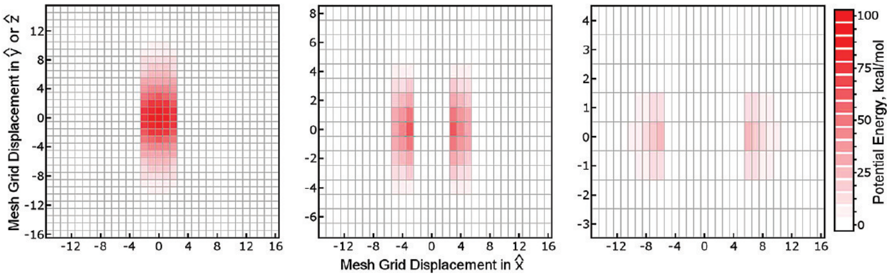

Figure 3. A three-level MLE scheme. As illustrated above, the reciprocal space pair potential mesh can be split into three (or more) separate meshes, each with successively larger coarsening factors. Here, there are two coarsened meshes, with coarsening factors C_{yz} of 2 and 4, respectively. In this scheme, the pair potential in the lowest level mesh extends 2 pages; slabs of the lowest level charge mesh would require 4 pages of zero-padding. The pair potential in the intermediate level mesh has $T_{cut} =$ 5, although only 6 of its pages have nonzero potential values in them (the thickness of the nonzero region of the mesh is $2 \times 5 + 1 = 11$ pages). While slabs of the intermediate-level charge mesh would require 10 pages of zero-padding, the intermediate level mesh is much smaller than the lowest level mesh, making such a degree of padding more economical. The color scale is not the same as that for Figure 1 because the SPME calculation this MLE scheme approximates was not the same; the diagram is intended for qualitative understanding only.

highest level charge mesh can be 2−6 times larger than the mesh spacing in Q, calculating $Q_{c,L}★\theta_{c,L}^{(rec)}$ is not so demanding as calculating $Q★\theta^{(rec)}$, and the communication burden is likewise reduced.

2.3. Considerations for Constant Pressure Simulations. It is important to note that, to make MLE run efficiently, $\theta^{(rec)}$ must be computed by taking the inverse Fourier transform of $\mathscr{F}(\theta^{(rec)})$ as is typically computed in particle-mesh methods, then extracting $\theta_h^{(rec)}$ and $\theta_c^{(rec)}$, and finally computing the Fourier transforms $\mathscr{F}(\theta_h^{(rec)})$ and $\mathscr{F}(\theta_c^{(rec)})$. This preparatory work must be done at the beginning of the simulation so that during each step of dynamics the necessary convolutions described in eq 12 or 13 can be accomplished with only two Fourier transforms each. This necessity may appear to limit the applicability of the MLE approximation to constant volume systems, where $\theta^{(rec)}$ is constant throughout the simulation. However, if the unit cell volume rescaling in constant pressure simulations is isotropic, and the Gaussian charge smoothing parameter σ and the mesh spacing μ vary in proportion to the unit cell dimensions, the updated pair potentials $\theta_h^{(rec)*}$ and $\theta_c^{(rec)*}$ for any new unit cell volume can be obtained be simply rescaling the $\theta_h^{(rec)}$ and $\theta_c^{(rec)}$ obtained at the beginning of the simulation.

Table 1. Test Cases for the Multi-Level Ewald Method[a]

case	cell dimensions (*a*, *b*, *c*), Å	cell dimensions (α, *β*, *γ*)	atom count
water	31.4 × 31.4 × 31.4	90°, 90°, 90°	3072
streptavidin	89.7 × 89.7 × 89.7	90°, 90°, 90°	73 305
protein crystal	71.4 × 71.4 × 75.6	90°, 90°, 120°	36 414
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.

3. Methods

3.1. The MDGX Program. To test the Multi-Level Ewald (MLE) method, we wrote an in-house molecular dynamics program, MDGX (Molecular Dynamics with Gaussian Charges and Explicit Polarization—not all parts of the acronym are yet fulfilled, as the purpose of the program is to be a proving ground for new algorithms). Routines in MDGX are able to read AMBER topology files and produce outputs in a format like that of the SANDER module in the AMBER software package.[19] The MDGX program is able to run unconstrained molecular dynamics trajectories of systems such as a box of SPC-Fw water molecules[20] in the microcanonical (NVE) ensemble or simply compute energies and forces acting on all atoms of a system for a single set of coordinates. The MDGX program implements both Smooth Particle-Mesh Ewald (SPME) as well as our new MLE method and also offers the option of using different particle ⇋ mesh interpolation orders in different dimensions, a feature that we will show is very helpful for tuning MLE. When run with identical parameters, the SPME reciprocal space electrostatic forces computed by MDGX agree with those of sander to 1.0×10^{-9} relative precision. MDGX links with the FFTW[21] library to perform its FFTs.

3.2. A Matlab Ewald Calculator. While the MDGX program is an excellent tool for testing MLE and other new variants of P^3M, it currently only works with orthorhombic unit cells (the reciprocal space code is actually set up to perform calculations with nonorthorhombic cells, but the direct space domain decomposition is not yet ready in this respect). The MDGX program was therefore only used for calculations involving rectangular unit cells.

Before MDGX was created, Multi-Level Ewald was discovered and verified through a set of script functions written for the Matlab software package (The MathWorks, Inc., Natick, MA). These scripts, which can easily produce forces and energies for a particular set of coordinates and charges but are not efficient enough to propagate a lengthy molecular dynamics trajectory, were used for any calculations involving nonorthorhombic unit cells. The calculator facilitates analysis of every stage of the Multi-Level Ewald process through Matlab's high-level programming language and is available from the authors on request.

3.3. Test Systems. As presented in Table 1, we chose a number of systems representative of those found in typical, condensed phase, biomolecular simulations. The first, a system of 1024 SPC-Fw water molecules, was used for testing energy conservation and ensemble properties of the system collected over long molecular dynamics runs. The other systems were much larger protein-in-water and protein crystal systems used in our previous study developing a different P^3M method.[22] Most importantly, these systems span a variety of unit cell types: as will be shown, MLE can be performed with any type of unit cell, but the geometry of the unit cell itself affects the accuracy of the MLE approximation.

3.4. Smooth Particle-Mesh Ewald Accuracy Standards and Reference Calculations. Standards for the accuracy of electrostatic forces in molecular simulations must be established before assessing the accuracy of MLE approximations with respect to SPME targets. Generally, we chose the accuracy obtained by the default settings of the SANDER molecular dynamics engine in the AMBER software package[19] as a reasonable level of accuracy for molecular simulations. These settings are mesh spacing μ as close to 1.0 Å as possible given a mesh size g with prime factors 2, 3, 5, and possibly 7, particle ⇋ mesh interpolation order $I^{(pm)} = 4$, and direct sum tolerance $D_{tol} = 1.0 \times 10^{-5}$ with direct space cutoff $L_{cut} = 8.0$ Å leading to a Gaussian charge smoothing half width $\sigma = 1.434$ Å. They can be expected to produce electrostatic forces with a root mean squared (rms) error of about 1.0×10^{-2} kcal/mol·Å, but the exact number varies depending on the system composition and geometry. Most SPME calculations for this work were performed with these parameters, and most modifications to the parameters were made in such a way as to conserve the overall accuracy of the calculation. For reference, very high-quality SPME calculations were performed with $\mu \sim 0.4$ Å, $I^{(pm)} = 8$, and an identical value of σ to ensure that the direct space and reciprocal space components of the SPME calculation could both be compared to the reference. The reference calculations produced forces convergent to within 1.0×10^{-6} kcal/mol·Å.

4. Results

4.1. Accuracy of Forces and Energies Computed with Multi-Level Ewald. The most important products of our new Ewald reciprocal space approximation are correct reproduction of the electrostatic energy of a system of particles and correct reproduction of the gradients of that energy. As our implementation of Multi-Level Ewald (MLE) is an approximation to the Smooth Particle-Mesh Ewald (SPME) method, we performed SPME calculations with high-accuracy "reference" parameters as well as typical molecular dynamics parameters and compared MLE results to both. In typical SPME calculations, there are two sources of error to consider, arising from the direct and reciprocal space parts of the calculation, respectively. MLE uses the same direct space sum but approximates the reciprocal space sum, introducing "coarsening" errors into the electrostatics calculation. We define the coarsening errors as deviations in the MLE approximation away from the equivalent SPME calculation, where the "equivalent" SPME calculation uses the same σ, $I^{(pm)}$, and μ parameters as the MLE calculation and has its own degree of inaccuracy relative to the SPME reference calculation. Fundamentally, the coarsening errors are errors in the scalar values of the electrostatic potential

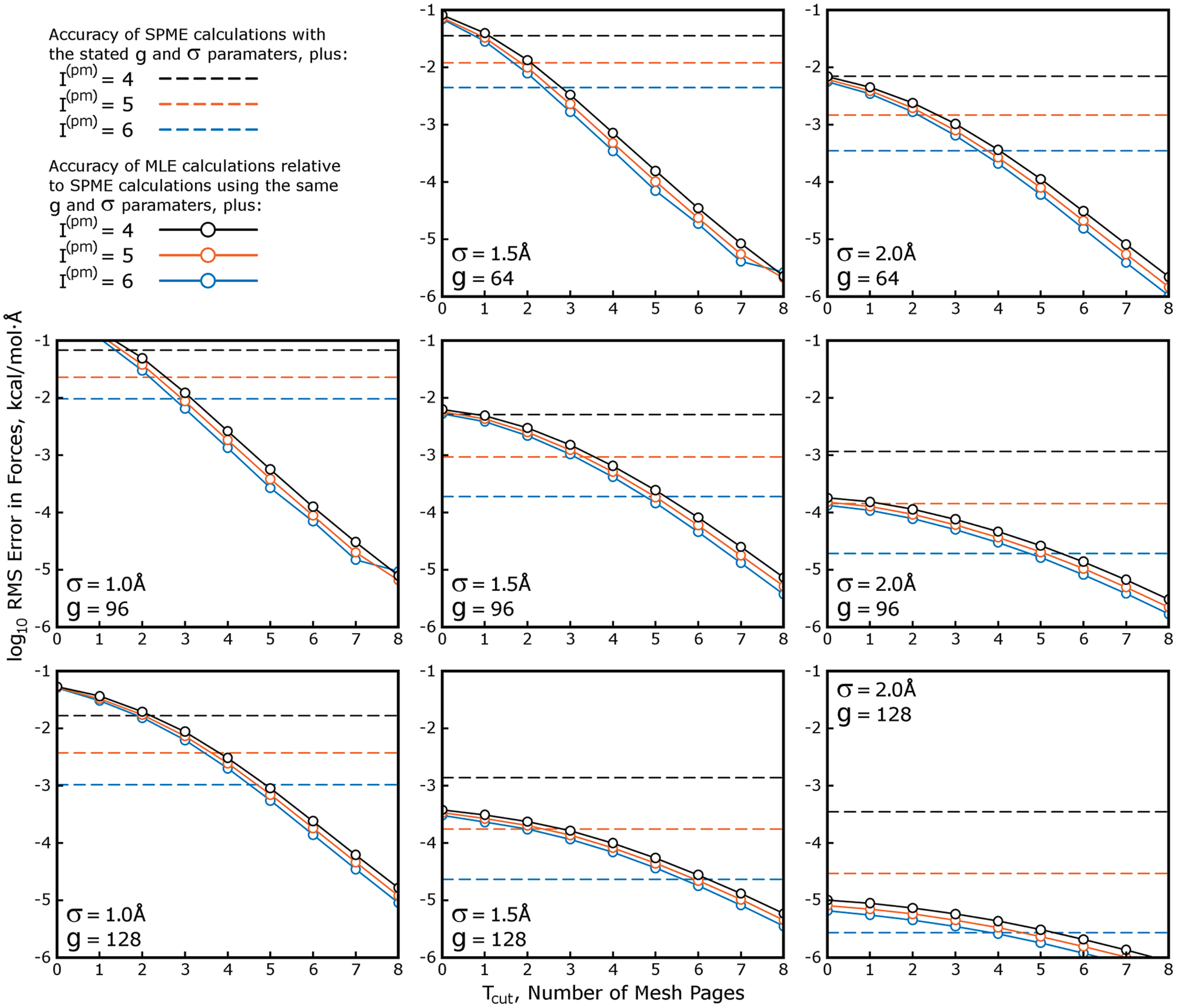

Figure 4. Accuracy of MLE calculations shows dependence on the parameters of the SPME calculation that is being approximated. SPME calculations were performed on the streptavidin test case with a range of Gaussian charge smoothing half widths σ and mesh sizes $g \times g \times g$. In practice, SPME calculations run with particle ⇋ mesh interpolation order $I^{(pm)} = 4$ require that σ be at least 1.5× the mesh spacing μ to produce reasonable accuracy in the forces arising from the reciprocal space part; however, if $I^{(pm)}$ is set to 6, σ:μ ratios as small as 1.0 can be used. The σ:μ ratio in the center panel is roughly 1.6, and it increases moving across the panels from left to right or top to bottom. The $\sigma = 1.0$ Å, $g = 64$ case is omitted because it would be far too inaccurate for molecular simulations, no matter the value of $I^{(pm)}$. In each panel, horizontal dashed lines show the accuracy of SPME calculations with the stated parameters (the "equivalent" SPME calculations) relative to a high-accuracy reference calculation performed as described in Methods. Solid lines with "○" depict the accuracy of MLE calculations relative to the equivalent SPME calculations.

at points in $U^{(rec)}$ (see eq 6), which in turn imply errors in the forces on charged particles interpolated from $U^{(rec)}$. We will first focus on the errors in forces, as these are of greatest importance in molecular simulations; energies will be discussed in another section.

To quantify the coarsening errors as a function of the SPME parameters, we ran calculations on the Streptavidin test case described in Table 1 using a range of values for the SPME mesh spacing μ, Gaussian charge smoothing function half width σ, and interpolation order $I^{(pm)}$. We then approximated the SPME results with MLE calculations using $C_{yz} = 2$, mesh ⇋ mesh interpolation order $I^{(mm)} = 8$, and a range of values for T_{cut} (C_{yz} and $I^{(mm)}$ can be varied to benefit the accuracy of MLE calculations, as we will show later, but their values were fixed for simplicity in this test). The results in Figure 4 show that the accuracy of the MLE approximation improves exponentially with T_{cut} and is also very sensitive to the parameters of the equivalent SPME calculation, particularly σ and μ and to a lesser extent $I^{(pm)}$. Although the values of μ and σ are widely varied and not thoroughly sampled, Figure 4 establishes another important result, that MLE can be used to approximate a wide range of different SPME calculations and, without large values of T_{cut}, incur less error than the SPME calculation itself.

Although at first it appears that the accuracy of MLE is least sensitive to $I^{(pm)}$, this parameter can be manipulated to great advantage in MLE calculations. Of all of the commercially or academically available molecular dynamics codes, the Desmond software package[23] is, to our knowledge, the only one to permit different settings of $I^{(pm)}$ in different

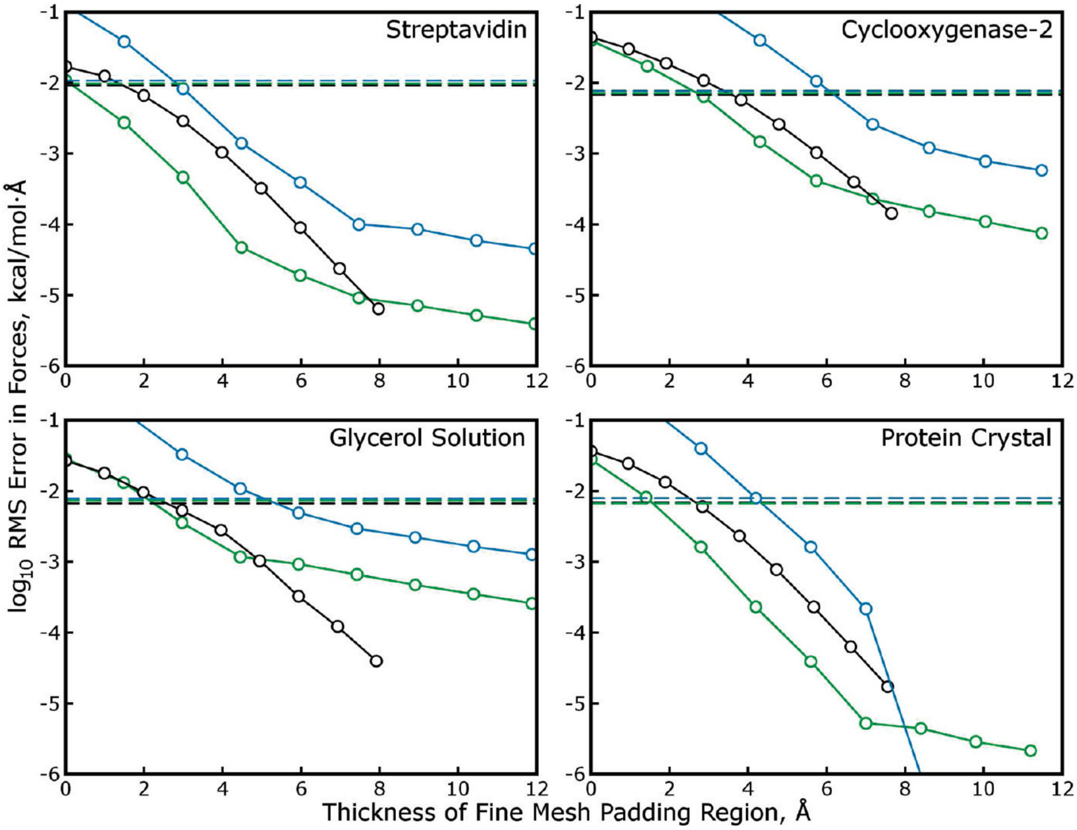

Figure 5. Anisotropic mesh spacings and interpolation orders enhance the accuracy of MLE calculations. SPME calculations were performed on four of the test cases from Table 1, this time using the AMBER default parameters $\sigma \approx 1.4$ Å, $I_x^{(pm)} = I_y^{(pm)} = I_z^{(pm)} = 4$, and the smallest mesh dimensions g_x, g_y, and g_z such that the g_x, g_y, and g_z were multiples of 2, 3, and 5 and the mesh spacings μ_x, μ_y, and μ_z were less than 1.0 Å. Accurate MLE approximations of such SPME calculations are possible for all of these systems, which include monoclinic and triclinic unit cells in addition to the cubic streptavidin system. Modifying the SPME parameters by increasing μ_x to 1.5 Å and increasing $I^{(pm)}$ to compensate maintains the accuracy of the SPME calculation with a smaller amount of mesh data and can also increase the accuracy of MLE approximations. As in Figure 4, dashed lines represent the accuracy of SPME calculations relative to a high accuracy reference, and lines with "○" represent the accuracy of MLE calculations relative to SPME. Black, green, and blue lines correspond to $I^{(pm)}$ = (4,4,4), (6,4,4), and (6,6,6), respectively. Because the SPME/MLE calculations in each panel use different values of μ_x, the results are plotted in terms of the physical thickness of the padding needed for each MLE approximation, $T_{cut} \times \mu$.

directions. However, we found that this is a powerful way to improve the accuracy of an MLE approximation. As we showed in previous work,[22] setting $I^{(pm)} = 6$ permits μ to be set as much as 1.5× larger than $I^{(pm)} = 4$ would allow; the result in fact applies in one, two, or all three dimensions. Strictly in terms of the number of operations, increasing $I^{(pm)}$ to 6 in only one dimension offers the most reduction in the mesh size per increase in the amount of particle ⇋ mesh work. For example, a mesh of 90^3 points could be replaced by a mesh of $60 \times 90 \times 90$ points, at the expense of mapping particles to $6 \times 4 \times 4 = 96$ points rather than $4 \times 4 \times 4 = 64$. In contrast, setting $I^{(pm)} = 6$ in all dimensions could produce comparable accuracy in the aforementioned problem with a mesh of 60^3 points, but at the expense of mapping all particles to 216 mesh points.

We performed additional SPME and MLE calculations on the streptavidin system, this time using the AMBER default parameters (as described in Methods) and a variation on those parameters using $I^{(pm)} = 6$ and μ approaching 1.5 Å in the $\hat{x}$ direction or in all directions. We also performed tests on other systems described in Table 1 to confirm the accuracy of MLE when applied to nonorthorhombic unit cells. All of these results are presented in Figure 5. While all of the different combinations of $I^{(pm)}$ and μ produce comparable accuracy in the SPME calculation, and while raising $I^{(pm)}$ will improve the accuracy of MLE calculations if all μ (and σ) are held fixed, increasing μ in this manner appears to be detrimental to the accuracy of the subsequent MLE approximation. However, if only $I_x^{(pm)}$ is raised and μ_x is increased accordingly, the accuracy of the subsequent MLE approximation is improved significantly in three out of the four cases. Anisotropic interpolation orders and a longer mesh spacing in the $\hat{x}$ direction therefore permit significant reductions in the number of pages T_{cut} that must be computed in zero-padded FFTs and transmitted between neighboring slabs, making MLE cheaper to apply.

As can be seen in Figure 4, raising $I^{(pm)}$ is not the only way to compensate for an increase in μ. Raising σ itself is another way to maintain the critical ratio of σ to μ. Larger values of σ are obtained by using a longer direct space cutoff L_{cut}; many codes[23,24] and specialized hardware for running molecular simulations[25] make use of longer values of L_{cut} to reduce the size of the reciprocal space mesh. We therefore tested the accuracy of MLE calculations if larger values of

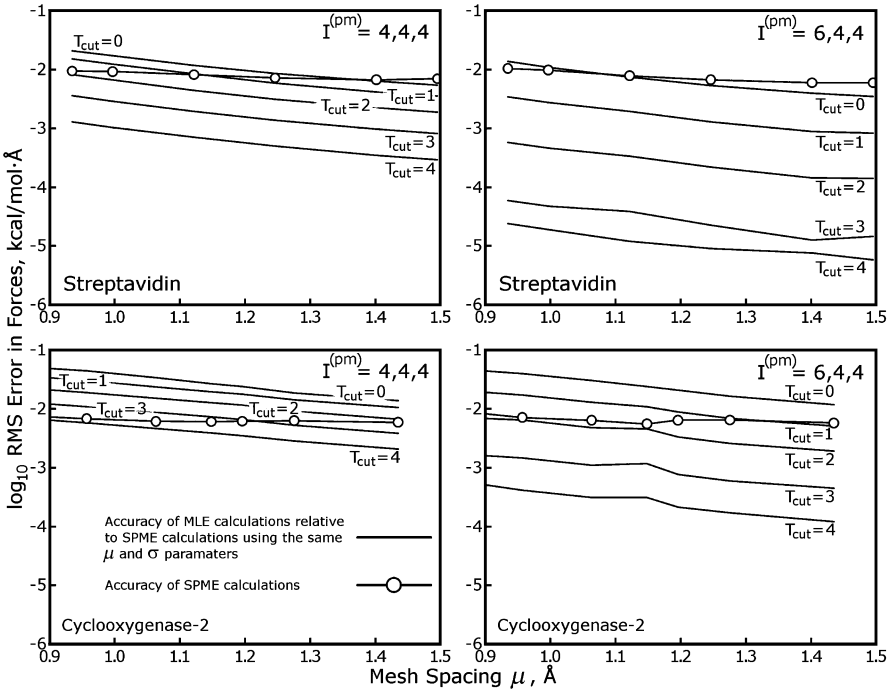

Figure 6. Wider Gaussian smoothing functions enhance the accuracy of MLE. Although using a larger mesh spacing μ in conjunction with isotropic 6th order particle ⇋ mesh interpolation is detrimental to the accuracy of MLE approximations, it is possible to improve the accuracy of MLE by using a larger μ and increasing σ, the rms of the Gaussian charge smoothing function, to maintain the accuracy of the equivalent SPME calculation. The accuracy of MLE approximations for the streptavidin and cyclooxygenase-2 systems is plotted as a function of μ. For each of the equivalent SPME calculations, σ was adjusted in proportion to μ to maintain the σ to μ ratio that would be obtained in each system by the AMBER default parameters, roughly 1.42. As shown by the solid lines with "○", this approach is also effective at conserving the accuracy of the equivalent SPME calculation, even improving it slightly as μ gets larger. The accuracies of MLE approximations improve steadily as a function of μ. The inset legend in the lower left panel applies to all panels.

σ, rather than higher $I^{(pm)}$, were used in conjunction with a larger μ. The results in Figure 6 stand in contrast to the results of Figure 5: the MLE approximation becomes more accurate when longer μ are used, insofar as σ is increased accordingly. When using higher σ and larger μ, anisotropic particle ⇋ mesh interpolation is still effective at conserving the accuracy of the SPME calculation and continues to benefit the accuracy of the MLE approximation.

Noting that nonorthorhombic unit cells are detrimental to the accuracy of MLE (although only to the extent that T_{cut} must be raised by 1 or 2), we tried MLE calculations with several other monoclinic unit cells, each with only one of the α, β, or γ angles different from 90°. While we had hoped that MLE might be able to give the same accuracy in monoclinic unit cells as in orthorhombic ones if the coarsening occurred in certain dimensions with respect to the nonorthogonal unit cell vectors, the accuracy of MLE showed similar degradation no matter which angle differed from 90° (data not shown).

As mentioned in the Theory, the MLE approximation is tunable in the $I^{(mm)}$ parameter as well as in T_{cut}. Knowing that high values of $I^{(mm)}$ are economical in terms of the number of arithmetic operations, we tested the accuracy of MLE for orders of mesh ⇋ mesh interpolation ranging from 4 to 16. The AMBER default parameters and variants with $I^{(pm)} = 6$ were again used for this test. As shown in Figure 7, if a low order of mesh ⇋ mesh interpolation can produce accuracy on the order of the SPME reciprocal space calculation, raising $I^{(mm)}$ can improve the accuracy of an MLE approximation by an order of magnitude.

Figures 5 and 7 show that, with proper choices of T_{cut} and $I^{(mm)}$ to accommodate the parameters of the equivalent SPME calculation, the coarsening errors in MLE calculations can be well below the level of reciprocal space error in the equivalent SPME calculation. However, the form of the coarsening errors themselves must be examined. As will be discussed in the following section, the reciprocal space electrostatic forces accumulate errors as a consequence of inaccuracies in the mesh $U^{(rec)}$, but because the interpolation of $U^{(rec)}$ itself is done only along certain dimensions in MLE calculations, the resulting errors in the reciprocal space electrostatic forces could be expected to be somewhat anisotropic. Figure 8 shows the magnitudes of the coarsening errors acting on individual atoms of the streptavidin system in the $\hat{x}$, $\hat{y}$, and $\hat{z}$ directions under an aggressive MLE approximation (details are given in the figure itself). As might be expected, the coarsening errors tend to be greater in the $\hat{y}$ and $\hat{z}$ directions, but only slightly: coarsening errors in the electrostatic forces also have significant components in the $\hat{x}$ direction, despite the fact that no mesh coarsening was done in $\hat{x}$. As was shown in Figure 5, use of anisotropic SPME parameters reduces the overall error in the MLE approximation, but the individual errors in forces become even more shifted toward the $\hat{y}$ and $\hat{z}$ directions. Conceptually, anisotropic errors are less desirable than isotropic ones, as they might impart the wrong energetics to interactions

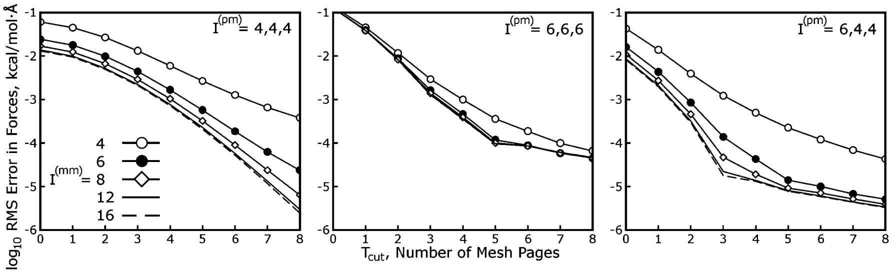

Figure 7. Higher mesh $\leftrightharpoons$ mesh interpolation orders can benefit MLE calculations. Interpolation between the finest mesh and higher level meshes in MLE is, like the particle $\leftrightharpoons$ mesh interpolation in standard SPME calculations, based on cardinal B-splines; the grid points of the finest mesh can be thought of as particles to map onto coarser meshes. However, mesh $\leftrightharpoons$ mesh interpolation of order $I^{(mm)}$ only occurs in only two dimensions, and, because of the regularity of the fine grid, the operations are separable in each dimension leading to $O(I^{(mm)})$ complexity and the possibly much higher orders of $I^{(mm)}$ than $I^{(pm)}$. The accuracy of MLE approximations to the SPME calculations described in Figure 5 was therefore re-evaluated as a function of $I^{(mm)}$ for the streptavidin test case. The results suggest that raising $I^{(mm)}$ is effective if the equivalent SPME calculation makes use of a high σ:μ ratio in the dimensions along which the mesh is coarsened (i.e., $\hat{y}$ and $\hat{z}$). The inset legend in the leftmost panel applies to all panels.

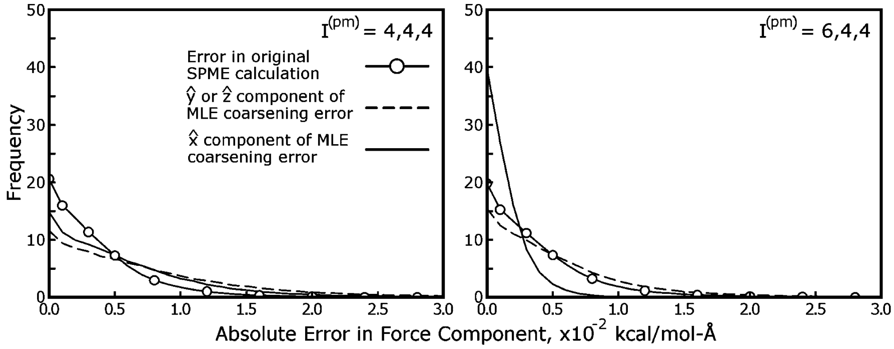

Figure 8. Errors arising from the MLE approximation are anisotropic. Because the MLE approximation is applied in only two of the three unit cell dimensions, errors arising from the approximation may be larger in some dimensions than others. Electrostatic forces on atoms of the streptavidin system in Table 1 were computed using SPME and the AMBER default parameters with the particle $\leftrightharpoons$ mesh interpolation schemes given in each panel. As shown by these histograms, the coarsening errors (differences between the equivalent SPME calculation and an aggressive MLE approximation with $T_{cut} = 0$, $C_{yz} = 2$, and $I^{(mm)} = 8$) are indeed more pronounced in the directions along which the reciprocal space mesh was coarsened, $\hat{y}$ and $\hat{z}$, even if the equivalent SPME calculation uses isotropic μ and σ parameters. Increasing μ_x to ~1.5 Å and setting $I_x^{(pm)} = 6$ reduces errors in all directions, but the accuracy of MLE-approximated forces in the $\hat{x}$ direction shows the most improvement by far, even exceeding the accuracy of the equivalent SPME calculation. The frequencies of errors in the $\hat{y}$ and $\hat{z}$ directions were averaged and presented together as they were indistinguishable in this cubic unit cell. For comparison, we also show a histogram of the magnitudes of errors inherent in the equivalent SPME reciprocal space calculation, as judged by a high accuracy standard. The inset legend in the leftmost panel applies to both panels.

based on, for example, the orientation of a protein in the simulation cell. However, we stress that this test used an aggressive MLE approximation for demonstrative purposes and that other MLE parameters can yield errors well below those of the equivalent SPME calculation. We attempted the same test with more conservative parameters (data not shown) and found the shapes of the force error histograms to be very similar to those in Figure 5, but on an exponentially smaller scale.

4.2. Larger Coarsening Factors and Incorporation of Multiple Higher-Level Meshes. While there may be some advantage in being able to split the convolution $Q\bigstar\theta^{(rec)}$ into multiple pieces (and, if T_{cut} can be set as low as 0, obtain $Q_h\bigstar\theta_h^{(rec)}$ by performing only two-dimensional FFTs, saving some FFT work and a major data transpose operation), C_{yz} can be larger than 2 to reduce the size and processing requirements of the $Q_c\bigstar\theta_c^{(rec)}$ convolution even further. The MLE scheme is also not limited to just one higher level mesh: the charge mesh Q can be split into a series of meshes $Q_{c,1}$, $Q_{c,2}$, ..., $Q_{c,L}$, staged with increasing values of $C_{yz,1}$, $C_{yz,2}$, ..., $C_{yz,L}$ depending on the size of the problem.

Figure 9 shows the accuracy of Multi-Level Ewald on two of the systems in Table 1 using larger values of C_{yz}, demonstrating that MLE can be applied with C_{yz} as high as 4−6 for 16−36-fold reductions in the amount of data present in the coarsest mesh. However, setting T_{cut} to 4 or 6 could be very expensive in terms of the extra FFT work and communication cost. If eight MLE slabs were used with a mesh of 64 × 96 × 96 points with C_{yz} set to 4 and T_{cut} set to 5, each MLE slab would measure $(64/8) + 2 \times 5 = 18$ points thick; the FFT work needed to compute $Q_c\bigstar\theta_c^{(rec)}$ would be more than 16 times less than that needed to compute $Q\bigstar\theta^{(rec)}$ in the equivalent SPME calculation, but the FFT work needed to compute the series $\sum_i Q_i\bigstar\theta_h^{(rec)}$ would be roughly twice the original FFT burden. There would also

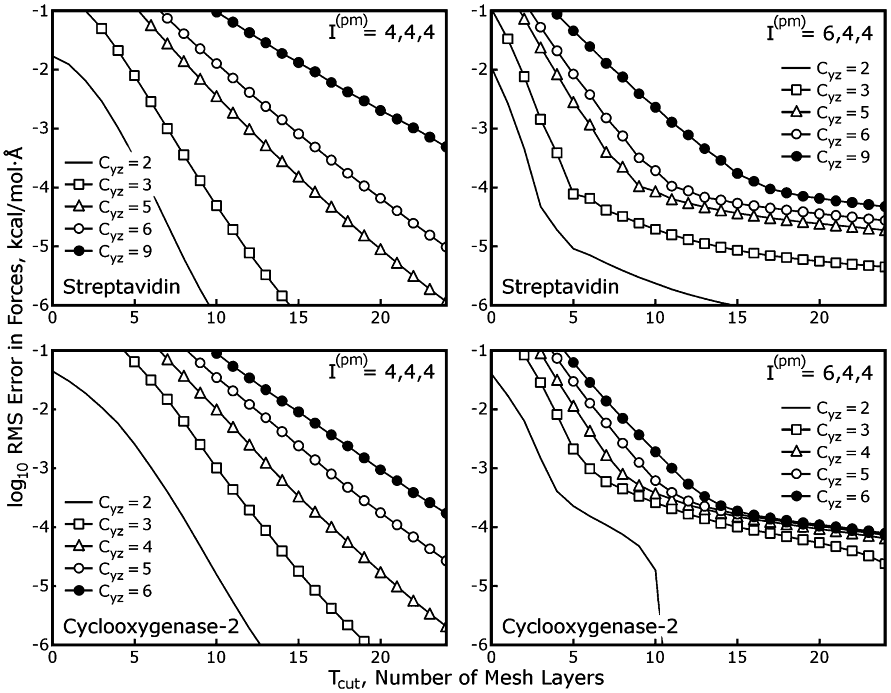

Figure 9. Larger coarsening factors are available in MLE. Thus far, the results have focused on the performance of the MLE approximation for a variety of SPME calculations, emphasizing what values of $I^{(mm)}$ and T_{cut} are necessary to achieve accurate results with a coarsening factor C_{yz} of 2, but C_{yz} is itself a tunable parameter of MLE. These plots show the accuracy of the MLE approximation for the streptavidin and cyclooxygenase-2 test cases (in cubic and triclinic unit cells, respectively) for numerous coarsening factors as shown in each diagram. The coarsening factors are limited to common factors of the mesh sizes in the $\hat{y}$ and $\hat{z}$ directions, but we do not expect this to be a serious limitation in practice. In these tests, $I^{(mm)}$ was fixed at 8. The AMBER default SPME paramaters, or the variant with anisotropic interpolation discussed in previous figures and the maint text, were used for the SPME calculations as indicated in each diagram. While larger values of C_{yz} require larger values of T_{cut} to produce accurate results, MLE with C_{yz} as high as 6 can imply modest additional error with T_{cut} as low as 7 if anisotropic particle $\leftrightharpoons$ mesh interpolation is used. As shown in Table 2, a third, intermediate mesh level, typically with $C_{yz} = 2$, is helpful for reducing the computation and communication burden of larger values of T_{cut}, making it possible to efficiently coarsen the reciprocal space mesh in stages.

be a considerable burden for communicating the zero-padded regions of each MLE slab.

To bridge the gap between Q and Q_c, we introduced another mesh with an intermediate coarsening factor (i.e., $C_{yz,2} = 2$). Following the nomenclature in the Theory, we will refer to this intermediate mesh as $Q_{c,2}$ and refer to the highest level mesh, coarsened by a high value of $C_{yz,3}$, as $Q_{c,3}$. Previously, we have used T_{cut} to describe the extent of the reciprocal space pair potential applied to the finest mesh Q or the number of zero-padded pages in each of its slabs. When multiple coarse meshes are involved, we refer to the extent of the potential for the nth mesh as $T_{cut,\,n}$ and the coarsening factor for the nth mesh as $C_{yz,\,n}$. (In principle, for the lowest level mesh, $C_{yz,1} = 1$, and for the highest level mesh $T_{cut,\,L}$ is not defined.) In a three-mesh scheme, convoluting the lowest and intermediate level meshes Q and $Q_{c,2}$ with an intermediate-ranged pair potential $\theta_{c,2}^{(rec)}$ can be accomplished as a series of convolutions over slabs as was done for Q in previous MLE calculations. The slabs of the intermediate coarsened mesh, much less dense than Q, could be padded by a high value of $T_{cut,2}$ without adding greatly to the overall FFT computation or communication burden.

The MDGX program, but not the Matlab MLE calculator, was written to accommodate more than one level of mesh coarsening. We therefore tested the accuracy of MLE with several three-level mesh schemes on the cubic streptavidin system, as shown in Table 2. The performance of MLE in these three-level schemes is almost exactly what would be expected if the errors associated with separate two-level MLE calculations using the same parameters were combined.

4.3. Energy Conservation and Equilibrium Properties in Simulations with Multi-Level Ewald. The accuracy of forces obtained by the MLE approximation is encouraging, but we must still test whether the type of errors introduced by MLE, which are of a different nature than the errors in direct or reciprocal space forces arising from a standard SPME calculation, are possibly detrimental in the context of simulations. We therefore used the MDGX program to simulate a system of 1024 SPC-Fw water molecules in the microcanonical ensemble. Two different MLE schemes were used, as described in Table 3, both of them with three mesh levels. Trajectories were propagated at a 0.5 fs time step for 50 ns each, and the MLE or SPME reciprocal sums were computed at every time step to provide a stringent test of energy conservation. Coordinates were collected every 0.5 ps, and energies arising from electrostatic, Lennard-Jones, and harmonic bond and angle terms were collected every 0.05 ps.

The evolution of the total energy of the 1024 water system, simulated using each of the four methods described in Table

Table 2. Multi-Level Ewald Calculations Performed with Three Mesh Levels[a]

calc. type[b]	σ	$I_x^{(pm)}$	C_{yz}	T_{cut}	$\langle \Delta F^{(dir)} \rangle$[c]	$\langle \Delta F^{(rec)} \rangle$[d]	$\langle \Delta F^{(cor)} \rangle$[e]	$\langle \Delta F^{(ref)} \rangle$[f]
SPME	1.434	4			7.850	9.288		12.152
MLE	1.434	4	2	2	7.850	9.288	6.670	13.879
MLE	1.434	4	5	9	7.850	9.288	7.066	14.065
MLE	1.434	4	2,5	2,9	7.850	9.288	9.714	15.579
SPME	1.434	6			7.850	9.826		12.598
MLE	1.434	6	2	1	7.850	9.826	2.756	12.890
MLE	1.434	6	5	5	7.850	9.826	2.859	12.924
MLE	1.434	6	2,5	1,5	7.850	9.826	3.972	13.209
SPME	2.151	4			5.492	7.005		8.903
MLE	2.151	4	2	2	5.492	7.005	1.890	9.108
MLE	2.151	4	5	9	5.492	7.005	2.425	9.228
MLE	2.151	4	2,5	2,9	5.492	7.005	3.072	9.425

[a] All calculations in this table pertain to the streptavidin test case. Parameters for the equivalent SPME calculations included $I_y^{(pm)} = I_z^{(pm)} = 4$, $\sigma/\mu = 1.439$, $L_{cut} = 5.578\sigma$, and any parameters listed in the table pertaining to particular cases. $I^{(mm)}$ was set to 8 for all MLE calculations. The values of σ, μ, and L_{cut} in the first case are those obtained with the AMBER default settings for this system. All rms errors listed in this table are uncorrelated: when examined in detail, the Pearson correlation coefficients for the errors arising from distinct parts of the calculation are all smaller than 0.02. [b] Type of calculation. [c] The rms error in the direct space forces on all particles, relative to a high-accuracy SPME calculation ($\times 1.0 \times 10^{-3}$ kcal/mol·Å). [d] The rms error in the SPME reciprocal space force ($\times 1.0 \times 10^{-3}$ kcal/mol·Å). [e] The rms MLE coarsening error ($\times 1.0 \times 10^{-3}$ kcal/mol·Å). [f] Total rms error of the SPME or MLE calculation ($\times 1.0 \times 10^{-3}$ kcal/mol·Å).

Table 3. Parameters Used in Long-Time Scale Simulations of SPC-Fw Water, and the Accuracy of Forces Resulting from Each Approximation[a]

	conservative		aggressive	
parameter	SPME	MLE	SPME	MLE
L_{cut} (LJ, Å)[b]	10.0	10.0	10.0	10.0
L_{cut} (elec, Å)[c]	9.0	9.0	8.0	8.0
σ, Å	1.58	1.58	1.43	1.43
g	$(24 \times 36 \times 36)$	$(24 \times 36 \times 36)$	$(21 \times 32 \times 32)$	$(21 \times 32 \times 32)$
$I^{(pm)}$	$(6 \times 4 \times 4)$	$(6 \times 4 \times 4)$	$(6 \times 4 \times 4)$	$(6 \times 4 \times 4)$
$I^{(mm)}$		8		8
$C_{yz,2}$, $C_{yz,3}$		(2,4)		(2,4)
$T_{cut,1}$, $T_{cut,2}$		(1,5)		(0,4)
$\langle \Delta F^{(dir)} \rangle$[d]	4.9×10^{-3}	4.9×10^{-3}	7.7×10^{-3}	7.7×10^{-3}
$\langle \Delta F^{(rec)} \rangle$[e]	2.3×10^{-3}	2.5×10^{-3}	8.9×10^{-3}	1.3×10^{-2}

[a] The four simulations utilize either SPME or MLE calculations for long-ranged electrostatic interactions. While all simulations make use of the same direct space cutoffs, the "conservative" simulations use roughly 60% more data in Q for their SPME or MLE calculations. The "conservative" MLE scheme approximates the SPME results much more accurately than the equivalent SPME calculation obtains the true electrostatic force on each particle, as judged by a high-quality SPME calculation using $g = (96 \times 96 \times 96)$ and $I^{(pm)} = 8$. In contrast, the "aggressive" SPME scheme is somewhat less accurate, and the "aggressive" MLE scheme introduces roughly the same amount of error as the equivalent SPME scheme. Results from the simulations are presented in Figures 10 and 11. [b] Lennard-Jones potential truncation length. [c] Electrostatic direct space trucation length. [d] The rms error in direct space electrostatic forces with this approximation. [e] The rms error in reciprocal space electrostatic forces with this approximation (including MLE approximation, if applicable).

3, is shown in Figure 10. For comparison, the energy of the same system run using the "conservative" SPME parameters but a 1 fs time step is juxtaposed with the results at a 0.5 fs time step. While all of the 0.5 fs runs show some upward drift in the energy over 50 ns, it is very slow, and the time step itself clearly has a much greater impact on the energy conservation than the choice of SPME or MLE for a long-ranged electrostatics approximation. Neither MLE approximation, whether "aggressive" or "conservative", shows a visible difference when compared to the corresponding SPME calculation on the basis of energy conservation. Moreover, the differences between the two SPME methods are greater than the differences between the MLE approximations and the equivalent SPME calculations: while the axes in each panel of Figure 10 have similar scales, the energy of the system run with aggressive SPME and MLE parameters is somewhat higher and the fluctuation of the energy is noticeably larger. The reason for this increase in the recorded energy can be traced to the inaccuracies inherent in the SPME reciprocal space calculation when the σ to μ ratio becomes smaller, as explained in the Supporting Information for our previous work on Ewald sums.[22] When run with the same aggressive SPME parameters, the MLE approximation returns similar increases in the absolute energy and fluctuations in that energy; adding more aggressive MLE parameters on top of the lower-quality SPME method does not seem to affect the results much further.

As shown in Table 4, the bulk properties of the SPC-Fw water are not significantly perturbed by any of the SPME or MLE approximations. When taken in the context of a macroscopic observable such as the heat of vaporization, the differences between the total energy of the system measured by aggressive or conservative electrostatic parameters are negligible.

We also investigated the microscopic structure of the water when simulated with each approximation. One reason for choosing the flexible SPC-Fw water model was to test whether the MLE method, which produces its lowest accuracy when computing interactions between very nearby particles, could perturb the behavior of bonded atoms. (Although electrostatic interactions are excluded between bonded atoms in most molecular force fields, this exclusion is done by computing the interaction of two Gaussian-smoothed charges at the specified distance and subtracting this from the reciprocal space sum, which necessarily computes all interactions during the mesh convolution.) As shown in Figure 11, neither the MLE approximation nor the quality of the SPME method has any significant effect on either the oxygen:hydrogen bond length, the hydrogen: hydrogen distance within each water molecule, or the radial distributions of oxygen and hydrogen atoms on different water molecules.

The timings for these single-processor MLE runs also provide an indication of how much more computational effort MLE would require over the standard SPME method. The fact that simulating the 1024 water molecules requires only 6−10% longer with MLE than with SPME indicates that most codes with well-optimized FFT routines could implement MLE without much more computational effort than SPME. Notably, while the MLE schemes are more costly in terms of FFT work, the majority of the extra cost actually comes from the mesh ⇋ mesh interpolation. The FFTW

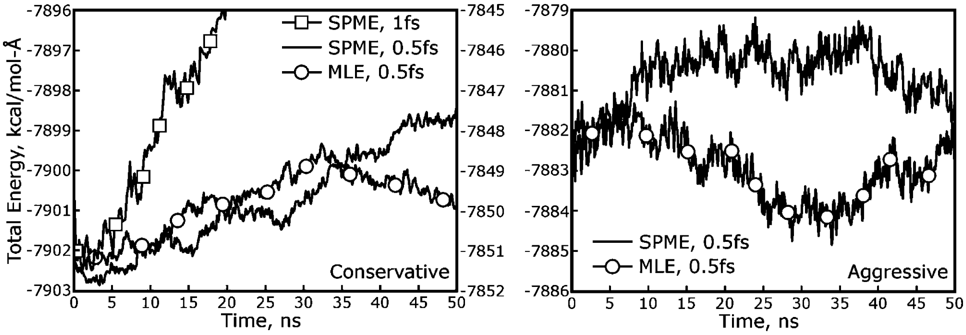

Figure 10. The Multi-Level Ewald approximation yields equivalent energy conservation to traditional Smooth Particle-Mesh Ewald. Simulations of a system of 1024 SPC-Fw water molecules show that MLE is able to conserve the system's energy over 50 ns trajectories. Parameters for each simulation are given in Table 3, and the length of the time step or style of Ewald summation is given in the legend of each figure. In each simulation, the total energy of the system fluctuates because both Ewald methods entail some degree of error as the particles move relative to the mesh, and the Lennard-Jones potential is sharply truncated at 10.5 Å. The total energy is therefore plotted as a series of mean values averaged over 200 frames each. Many investigators consider the energy conservation yielded by a 1 fs time step in systems with flexible bonds to hydrogen atoms acceptable. Comparison of the results obtained with a 1.0 fs time step (results should be read from the *y*-axis on the right side of the left-hand panel) to those obtained with a 0.5 fs time step shows that the time step itself can be a more significant contributor to the upward drift of the total system energy than many of the other parameters. (With the 0.5 fs time step, the temperature drift in each simulation is only about 0.5 K over 50 ns.) As the quality of the SPME calculations decreases, the fluctuation of the total system energy increases, as evident by comparing the results for SPME simulations in each panel. The MLE approximation also reproduces these changes in the sizes of fluctuations in the total energy.

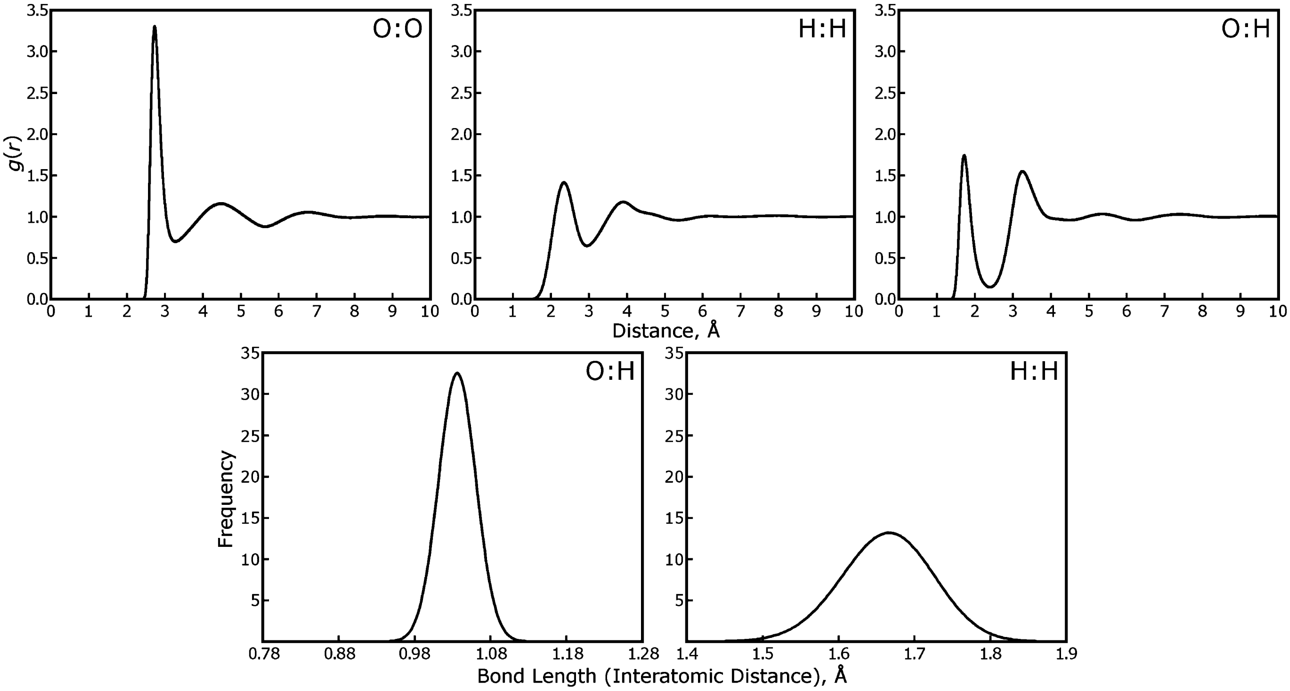

Figure 11. The average structure of SPC-Fw water molecules is maintained under different Ewald approximations. Analysis of the microscopic structure of the water molecules was performed to complement the energy conservation studies presented in Figure 10. Radial distribution functions for oxygen to oxygen, hydrogen to hydrogen, and oxygen to hydrogen atoms of SPC-Fw water molecules are displayed in the top three panels. Histrograms of the oxygen−hydrogen bond length and hydrogen−hydrogen intramolecular distance are shown in the lower two panels. Results from simulations with all four of the Ewald approximations listed in Table 3 are shown with solid black lines in each plot; the distributions overlap so precisely that they are indistinguishable.

libraries used by MDGX are among the fastest available, but as shown in Table 5, many other aspects of the MDGX code are not as efficient as their counterparts in PMEMD. (We are looking into compiler optimizations that may make the difference, as we believe we have coded routines such as the particle ⇋ mesh interpolation as efficiently as possible, and there appears to be little difference between the structure of our routines and those of PMEMD.) The estimates presented in Table 5 do not include the cost of computing FFTs over zero-padded regions of each MLE slab or the possible benefits of performing numerous FFTs over small regions rather than one large FFT; instead, a single convolu-

Table 4. Bulk Properties of 1024 Water Molecules Simulated in the Microcanonical Ensemble[a]

		conservative		aggressive	
property	PMEMD[e]	SPME	MLE	SPME	MLE
ΔH^{vap}, kcal/mol[b]	10.89 ± 0.00	10.88 ± 0.00	10.88 ± 0.00	10.88 ± 0.00	10.89 ± 0.00
T, K[c]	289.07 ± 0.08	289.59 ± 0.14	289.65 ± 0.03	291.72 ± 0.13	291.79 ± 0.17
D, $\times 10^{-5}$ cm ^{2}/s[d]	1.81 ± 0.02	1.83 ± 0.04	1.86 ± 0.05	1.85 ± 0.08	1.83 ± 0.07

[a] All values are given as averages over 12.5 ns blocks of each simulation, with standard deviations. [b] Heat of vaporization, calculated by $\Delta H^{vap} = -\langle E\rangle + RT$, where E is the mean potential energy, R is the gas constant, and T is the mean temperature. [c] Mean temperature of the simulation. [d] Diffusion coefficient. [e] Results obtained using the PMEMD implementation of the SANDER program from the AMBER software package, running SPME with the "Aggressive" parameters, but $l^{(pm)}$ set uniformly to 4 and g set uniformly to 32.

Table 5. Timings for MDGX or PMEMD on a Single Processor[a]

	conservative		aggressive		
routine	SPME	MLE	PMEMD	SPME	MLE
bonded interactions	5.8	5.9	4.6	5.8	5.8
$\Delta E^{(dir)}$, pair list	7.7	7.9	31.3	7.7	7.8
$\Delta E^{(dir)}$, interactions	383.9	383.6	216.7	323.8	323.9
$\Delta E^{(dir)}$, **total**	**391.6**	**391.6**	**248.0**	**331.5**	**331.7**
$\Delta E^{(rec)}$, B-splines	14.3	14.4	2.6	14.2	14.3
$\Delta E^{(rec)}$, particle → mesh	16.2	16.4	6.0	15.9	16.1
$\Delta E^{(rec)}$, convolution[b]	2.4	1.2	11.6	1.7	0.8
$\Delta E^{(rec)}$, FFT	17.3	22.6	28.4	9.1	11.8
$\Delta E^{(rec)}$, mesh → particle	22.5	22.6	11.1	22.2	22.3
mesh $\leftrightharpoons$ mesh[c]		42.1			29.0
$\Delta E^{(rec)}$, **total**	**72.8**	**119.3**	**59.8**	**63.1**	**94.2**
total wall time	**473.1**	**522.5**	**316.1**	**405.9**	**435.6**

[a] 20 000 steps of dynamics were run using PMEMD or MDGX in SPME or MLE mode on an Intel Q9550 processor (Core2 architecture, 6 MB L2 cache, 2.83 GHz clock speed). Simulations made use of the parameters in Table 3; the PMEMD simulation for the "aggressive" parameters used $l^{(pm)} = 4$ and $g = 32$. Timings for different categories of calculations in the MDGX code were measured using the UNIX gettimeofday() function; those in the PMEMD code were measured with PMEMD's internal profiling functions. Because the two programs are structured differently, the exact content of each category of calculation may not match exactly; for instance, MDGX and PMEMD use different styles of pair list, and the convolution kernel is computed more rapidly in MDGX than PMEMD because PMEMD always computes virial contributions while MDGX skips them if they are not needed. Total run times for both programs are greater than the sums of all categories in each column because timings for miscellaneous routines are not listed. Standard deviations were collected over 10 trials to estimate errors in the timings; they were less than 1% across all categories, but are omitted to condense the table. [b] Multiplication of the charge mesh and reciprocal space pair potential in Fourier space. [c] Interpolation between different levels of MLE mesh.

tion over a single mesh is done at all levels of the calculation in these single-processor runs. The estimates in Table 5 also neglect some possible benefits in the case of the "aggressive" MLE parameters. Setting $T_{\mathrm{cut},1} = 0$ permits convolutions of the lowest level mesh to be completed with two-dimensional FFTs, saving roughly 1/3 of the FFT work for that mesh level. We are continuing to develop the MDGX program to take advantage of these optimizations.

5. Discussion

5.1. Development of Multi-Level Ewald for Parallel Applications. We do not yet have a parallel version of MDGX to run Multi-Level Ewald on many processors. However, we believe that MLE can benefit massively parallel simulations, particularly when extremely powerful, multicore nodes must be connected by comparatively weak networks, when other forms of network heterogeneity are involved, or when the problem size is very large. With MLE, there are extra communication steps as the coarse meshes must be assembled and the electrostatic potential data deposited in the zero-padded "tails" of each slab must be passed to neighboring slabs. However, because these are all local effects, the number of messages that must be passed to create the coarse meshes and contribute $U_{\mathrm{c}}^{(\mathrm{rec})}$ to $U^{(\mathrm{rec})}$ is bounded, whereas the number of messages that must be passed in a convolution involving the whole P^{3}M reciprocal space mesh grows, at best, as the square root of the number of processors.[26] For example, distributing the convolution for a 60 × 90 × 90 mesh in the streptavidin test case over six multicore nodes would require each node to transmit at least 0.65 MB (megabytes) of data (if data are transmitted in 32-bit precision). With a three-level MLE scheme placing one fine mesh slab and one intermediate mesh slab on each node and setting $C_2 = 2$, $T_{\mathrm{cut},1} = 0$, $C_3 = 5$, $T_{\mathrm{cut},2} = 5$, the total volume of data transmission between nodes could be reduced nearly 4-fold, to 0.19 MB, in the convolution step.

Several challenges remain to implementing MLE in an efficient parallel code. The most obvious is load-balancing: MLE introduces another layer of complexity for scheduling the completion of coarse mesh convolutions, plus the associated mesh $\leftrightharpoons$ mesh interpolation. Another challenge is that, while MLE can be tuned to reduce data transmission between weakly connected processors, if many networked nodes must collaborate on each MLE slab, the original data communication problems resurface. Whereas each of K nodes must pass 4 × sqrt(K) messages in the original P^{3}M convolution, with MLE and P slabs with $K \gg P$ the number of messages is 4 × sqrt(K/P) + M (for nodes devoted to fine mesh calculations) or 4 × sqrt(K)/C + M (for K/C^2 nodes devoted to coarse mesh calculations), where M is a small constant for mesh $\leftrightharpoons$ mesh interpolation. One possible extension of the MLE method may be helpful for the case of many (multicore) nodes collaborating on each MLE slab: subdividing the convolutions over fine mesh pages into pencils using an analogous sharp, anisotropic splitting technique, and then applying a one-dimensional coarsening to meshes spanning each MLE slab.

5.2. Application to Multiple Time Step Algorithms. The different mesh levels in MLE calculations may be excellent candidates for updates at different time steps, particularly because the anisotropic splitting completely captures local changes to a molecular system's electrostatics in the lower charge mesh levels. In contrast, the highest level charge mesh requires the most communication between processors devoted to disparate regions of the simulation cell, to complete the global convolution. The novel splitting approach of MLE may create its own unique types of artifacts in such simulations, however. We have shown that the sum of contributions from all mesh levels can recover a smooth potential, but this may be perturbed if the electrostatic potential of each mesh level is updated at different times. With any multiple time step method there can be subtle resonances that affect the statistical properties of the system;[27] we intend to investigate the stability and efficiency of MLE with multiple time steps in the future.

5.3. Application to Systems With Two-Dimensional Periodicity. While periodic boundary conditions in three dimensions have been shown to be equivalent or superior to alternative boundary conditions for many condensed-phase biomolecular simulations,[28,29] there are classes of problems, notably membrane protein simulations,[30] that produce different results if periodicity is suppressed in one dimension. Regular Ewald methods are available for imposing two-dimensional periodicity,[31,32] but they are prohibitively expensive for systems of many thousands of atoms. For larger systems, a pseudo two-dimensional periodicity may be imposed by lengthening the simulation cell in one dimension, say $\hat{x}$, confining the system to the middle of the simulation cell along $\hat{x}$ by some stochastic boundary condition or, more directly, by a physical set of walls such as sheets of platinum atoms, and then running P^{3}M calculations as usual, with three-dimensional periodicity, on the extended system.[33] This approach has been further refined by adding an electrostatic field to counteract the net dipole of the system in $\hat{x}$,[34] mirroring the way in which modified potential functions and zero-padding have been used in plasma physics and astrophysical gravity calculations.[35]

The MLE method may be suitable for systems with two-dimensional periodicity, although membrane protein simulations run in the isothermal−isobaric (NPT) ensemble tend to require anisotropic system rescaling, which MLE cannot accommodate exactly. It is likely possible to extend the MLE method to work in such cases by storing a small array of precomputed solutions of each mesh potential with different unit cell ratios and thereafter interpolating the solution for any particular time step. Other, more general, solutions to the problem of isolated boundary conditions are again found in Adaptive P^{3}M methods[14−16,36] and the Multilevel Summation method.[37,38] In all of these approaches, the advantage for isolated boundary conditions in one or more dimensions is that only the coarse mesh must be evaluated in the zero-padded, empty regions of the simulation cell.

5.4. Diversity of Problem Decompositions for Future Machines. In conclusion, we have shown that, for pairwise potentials that decay as the inverse distance between particles, it is feasible to subdivide the convolution in particle $\leftrightharpoons$ mesh calculations sharply and anisotropically into many separate slabs without significantly adding to the overall cost of the calculation. This technique and the many smooth splitting approaches that already exist should be applicable to simulations on current and next-generation parallel computers, where hundreds to thousands of processors must collaborate to deliver longer simulation trajectories. It is worthwhile to develop a variety of these multilevel decompositions, as future supercomputers may come in novel architectures that offer huge advantages depending on the details of the parallel algorithm.

Acknowledgment. D.S.C. thanks Dr. Peter Freddolino, Professor Ross Walker, and Dr. David Hardy for helpful conversations. This work was funded by NIH grant RR12255.

References

(1) Henzler-Wildman, K. A.; Thai, V.; Lei, M.; Ott, M.; Wolf-Watz, M.; Fenn, T.; Pozharski, E.; Wilson, M. A.; Petsko, G. A.; Karplus, M.; Hübner, C. G.; Kern, D. Intrinsic motions along an enzymatic reaction trajectory. *Nature* **2007**, *450*, 838–844.

(2) 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 system. *Biochemistry* **2008**, *47*, 12065–12077.

(3) Maruthamuthu, V.; Schulten, K.; Leckband, D. Elasticity and rupture of a multi-domain cell adhesion complex. *Biophys. J.* **2009**, *96*, 3005–3014.

(4) Freddolino, P. L.; Liu, F.; Grubele, M.; Schulten, K. Ten-microsecond molecular dynamics simulation of a fast-folding WW domain. *Biophys. J.* **2008**, *94*, L75–L77.

(5) 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.

(6) 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.

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

(8) Pollock, E. L.; Glosli, J. Comments on P^{3}M, FMM, and the Ewald method for large periodic Coulombic systems. *Comput. Phys. Commun.* **1996**, *95*, 93–110.

(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) Brandt, A.; Lubrecht, A. A. Multilevel matrix multiplication and fast solution of integral equations. *J. Comput. Phys.* **1990**, *90*, 348–370.

(12) Skeel, R. D.; Tezcan, I.; Hardy, D. J. Multiple grid methods for classical molecular dynamics. *J. Comput. Chem.* **2002**, *23*, 672–684.

(13) Kurzak, J.; Pettitt, B. M. Massively parallel implementation of a fast multipole method for distributed memory machines. *J. Parallel Distr. Comp.* **2005**, *65*, 870–881.

(14) Thacker, R. J.; Couchman, H. M. P. A parallel adaptive P^{3}M code with hierarchical particle reordering. *Comput. Phys. Commun.* **2006**, *174*, 540–554.

(15) Merz, H.; Pen, U.; Trac, H. Towards optimal parallel PM N-body codes: PMFAST. *New Astron.* **2005**, *10*, 393–407.

(16) Couchman, H. M. P. Mesh-refined P^{3}M: A fast adaptive *N*-body algorithm. *Astrophys. J.* **1991**, *368*, L23–L26.

(17) Sagui, C.; Darden, T. A. P^{3}M and PME: A comparison of the two methods. In *Simulation and Theory of Electrostatic Interactions in Solution. Proceedings of the American Institute of Physics Conference, Sante Fe, NM, 1999*; Pratt, L. R., Hummer, G., Eds.; Springer: Secaucus, NJ, 2000; Vol. 492, pp 104−113.

(18) Schoenberg, I. J. *Cardinal Spline Interpolation*; Society for Industrial and Applied Mathematics; Philadelphia, PA, 1973.

(19) Case, D. A.; Cheatham, T. E., III; Darden, T. A.; Gohlke, H.; Luo, R.; Mer, K. M., Jr.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The AMBER biomolecular simulation programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.

(20) Wu, Y.; Tepper, H. L.; Voth, G. A. Flexible simple point-charge water model with improved liquid-state properties. *J. Chem. Phys.* **2006**, *124*, 024503.

(21) Frigo, M.; Johnson, S. G. The design and implementation of FFTW3. *Proc. IEEE* **2005**, *93*, 216–231.

(22) Cerutti, D. S.; Duke, R. E.; Lybrand, T. P. Staggered Mesh Ewald: An extension of the Smooth Particle Mesh Ewald method adding great versatility. *J. Chem. Theory Comput.* **2009**, *5*, 2322–2238.

(23) Bowers, K. J.; Chow, E.; Xu, H.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossv́ary, I.; Moraes, M. A.; Sacerdoti, F. D.; Salmon, J. K.; Shan, Y.; Shaw, D. E. Scalable algorithms for molecular dynamics simulations on commodity clusters. *Conference on High Performance Networking and Computing, Proceedings of the 2006 ACM/IEEE conference on Supercomputing, Tampa, FL, 2006*; Association for Computing Machinery: New York, NY, 2006; p 84.

(24) 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.

(25) Shaw, D. E.; Deneroff, M. M.; Dror, R. O.; Kuskin, J. S.; Larson, R. H.; Salmon, J. K.; Young, C.; Batson, B.; Bowers, K. J.; Chao, J. C.; Eastwood, M. P.; Gagliardo, J.; Grossman, J. P.; Ho, C. R.; Ierardi, D. J.; Kolossvary, I.; Klepeis, J. L.; Layman, T.; McLeavey, C.; Moraes, M. A.; Mueller, R.; Priest, E. C.; Shan, Y.; Spengler, J.; Theobald, M.; Towles, B.; Wang, S. C. Anton, a special-purpose machine for molecular dynamics simulation. *Special purpose to warehouse computers, Proceedings of the 34 th annual international symposium on computer architecture, San Diego, CA, May, 2007*; ACM: New York, NY, 2007; Vol. 1, pp 1−12.

(26) Sbalzarini, I. F.; Walther, J. H.; Bergdorf, M.; Hieber, S. E.; Kotsalis, E. M.; Koumoutsakos, P. PPM - A highly efficient parallel particle-mesh library for the simulation of continuum systems. *J. Comput. Phys.* **2006**, *215*, 566–588.

(27) Ma, Q.; Izaguirre, J. A.; Skeel, R. D. Nonlinear instability in multiple time stepping molecular dynamics. *Computational Sciences, Proceedings of the 2003 ACM symposium on applied computing, Melbourne, FL, March, 2003*; ACM: New York, NY, 2003; Vol. 1, pp 167−171.

(28) 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.

(29) 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. *J. Mol. Struct. (THEOCHEM)* **2005**, *760*, 91–98.

(30) Bostick, D.; Berkowitz, M. The implementation of slab geometry for membrane-channel molecular dynamics Simulations. *Biophys. J.* **2003**, *65*, 97–107.

(31) Parry, D. E. The electrostatic potential in the surface region of an ionic crystal. *Surf. Sci.* **1975**, *49*, 433.

(32) Kawata, M.; Mikami, M. Rapid calculation of two-dimensional Ewald summation. *Chem. Phys. Lett.* **2006**, *340*, 157–164.

(33) Spohr, E. Effect of electrostatic boundary conditions and system size on the interfacial properties of water and aqueous solutions. *J. Chem. Phys.* **1997**, *107*, 6342–6348.

(34) Yeh, I.-C.; Berkowitz, M. Ewald summation for systems with slab geometry. *J. Chem. Phys.* **1999**, *111*, 3155–3162.

(35) Gelato, S.; Chernoff, D. F.; Wasserman, I. An adaptive hierarchical particle-mesh code with isolated boundary conditions. *Astrophys. J.* **1997**, *480*, 115–131.

(36) Eastwood, J. W. The Block P^3 M algorithm. *Comput. Phys. Commun.* **2008**, *179*, 46–50.

(37) Brandt, A.; Ilyin, V.; Makedonska, N.; Suwan, I. Multilevel summation and Monte Carlo simulations. *J. Mol. Liq.* **2006**, *127*, 37–39.

(38) Hardy, D. J.; Stone, J. E.; Schulten, K. Multilevel summation of electrostatic potentials using graphics processing units. *Parallel Comput.* **2009**, *35*, 164–177.

CT900522G