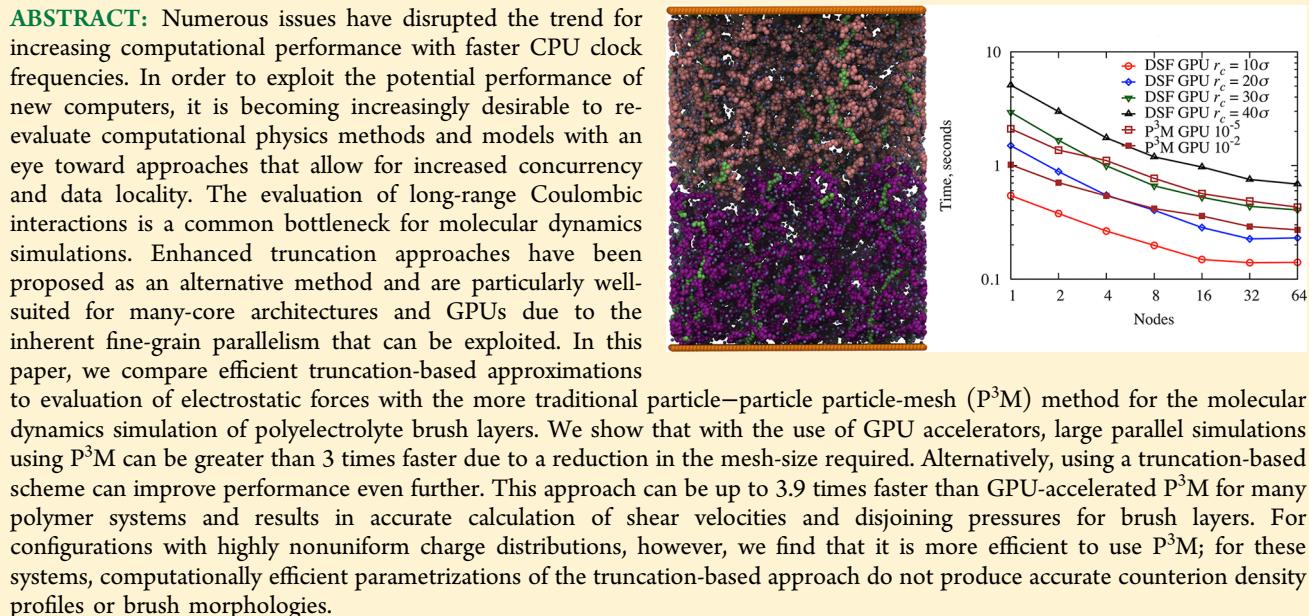


A Case Study of Truncated Electrostatics for Simulation of Polyelectrolyte Brushes on GPU Accelerators

Trung Dac Nguyen,[†] Jan-Michael Y. Carrillo,[‡] Andrey V. Dobrynin,[‡] and W. Michael Brown*,[†]

[†]National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, United States

[‡]Polymer Program, Institute of Materials Science and Department of Physics, University of Connecticut, Storrs, Connecticut 06269, United States



1. INTRODUCTION

Accurate calculation of electrostatic interactions is a fundamental requirement for many atomistic, molecular, and mesoscale simulations. The most common approaches for the calculation of electrostatic energies include Ewald summation and the related particle mesh methods.^{1–4} These methods separate the conditionally convergent electrostatic energy series into two absolutely convergent terms: a short-ranged part that is calculated within a certain cutoff in real space and a long-ranged part obtained by solving Poisson's equation in reciprocal space.¹ The best Ewald implementations have a time complexity of $O(N^{3/2})$ for N atoms. By representing the charge density on a mesh, the particle-mesh methods can achieve a time complexity of $O(M \log M)$, where M is the number of k-space mesh points ($M \approx N$ is typical). The most popular variants of these particle mesh methods include particle mesh Ewald (PME),² smooth particle mesh Ewald (SPME),³ and particle–particle particle-mesh (P^3M).⁴ Efficient particle-mesh implementations typically require a small fraction of the overall simulation time for serial molecular dynamics calculations. For large-scale simulations run on distributed memory supercomputers, however, they are a notorious bottleneck to performance due to the global communications required by

the reciprocal-space calculations.⁵ Although approaches to optimizing the reciprocal space calculation are an active area of research,^{6–8} methods to avoid the reciprocal space calculation altogether in order to achieve $O(N)$ time complexities with reduced global communications have also been proposed. Examples include, but are not limited to, multipole expansion, multilevel summation,⁹ reaction-field methods,^{10–12} and enhanced truncation methods.^{13–24} Depending on thermodynamic, structural, or dynamical features of interest, these alternative methods are shown to yield consistent results with those obtained using reciprocal-space methods.^{25–29}

Enhanced truncation methods focus the computational work for Coulombic interactions to summation within a spherical cutoff and correct for the lack of the long-range contribution by using a mean-field approximation^{13–15} or by incorporating image charges onto the surface of the cutoff sphere.^{17–19} Although these approaches can require larger cutoffs in order to obtain accurate results, there are several advantages. Enhanced truncation methods can achieve $O(N)$ time complexities, and their inherent locality offers superior parallel performance on

Received: August 15, 2012

Published: October 16, 2012

high-performance computers. Ewald and particle-mesh methods are only appropriate for periodically closed systems. In many cases, this is undesirable (e.g., interfacial systems), and the periodicity can lead to artifacts in the simulation. Approaches exist to circumvent these issues; however, they increase the computational time for the simulation and can introduce load-balancing issues for parallel simulation. Finally, exploiting the potential performance of GPU accelerators or other shared-memory many-core chips can be difficult with particle-mesh methods due to the relatively small mesh sizes and the overhead required to handle potential memory collisions with deterministic results.³⁰ Because both CPU and GPU accelerators increasingly require a greater degree of concurrency to achieve performance, performance gains for the entire simulation can be limited by particle-mesh methods due to Amdahl's law.

Recently, researchers have investigated the use of enhanced truncation approaches based on Wolf summation as an alternative to particle-mesh methods.^{18,19,24–29,31} Wolf et al. proposed evaluation of Coulombic interactions using a Coulombic potential truncated with a spherical cutoff and modified to enforce charge neutrality.¹⁷ The justification for the approach is that Coulombic interactions in condensed-phase systems are effectively short-ranged due to screening by other charged atoms. In order for the electrostatic sum to be absolutely convergent, charge neutralization within the cutoff radius is enforced by shifting the potential through placement of image charges on the cutoff sphere. Although this shifted form of the Coulombic potential has been shown to converge to the result obtained with the Ewald method,³² Wolf et al. found that including a damping term (the complementary error function used in the Ewald real-space term) improved the rate of convergence. The downside to this approach is that the potential converges to values that depend on the damping parameter.³²

Because the potential gradient proposed for force calculation with Wolf summation was not consistent with the potential (the potential is not differentiable at the cutoff), Zahn and co-workers proposed an alternative form suitable for molecular dynamics and demonstrated its accuracy in determining structural and dynamic properties of liquid TIP3P and SPC water systems.¹⁶ Since this potential is not continuous at the cutoff, Fennell and Gezelter proposed alternative forms to shift the force and energy consistently while ensuring smooth decay to zero at the cutoff.¹⁸ The so-called damped shifted force (DSF) potential for two particles with dimensionless charges q_i and q_j and a interatomic separation of r in reduced units (see section 2.1 below) is given by

$$U_{\text{DSF}}(r) = \begin{cases} q_i q_j \left[\frac{\text{erfc}(\alpha r)}{r^2} - \frac{\text{erfc}(\alpha r_c)}{r_c^2} + \left(\frac{\text{erfc}(\alpha r_c)}{r_c^2} + \frac{2\alpha}{\sqrt{\pi}} \frac{\exp(-\alpha^2 r_c^2)}{r_c} \right) (r - r_c) \right] & r < r_c \\ 0 & r \geq r_c \end{cases} \quad (1)$$

and the corresponding pair force

$$F_{\text{DSF}}(r) = \begin{cases} q_i q_j \left[\left(\frac{\text{erfc}(\alpha r)}{r^2} + \frac{2\alpha}{\sqrt{\pi}} \frac{\exp(-\alpha^2 r^2)}{r} \right) - \left(\frac{\text{erfc}(\alpha r_c)}{r_c^2} + \frac{2\alpha}{\sqrt{\pi}} \frac{\exp(-\alpha^2 r_c^2)}{r_c} \right) \right] & r < r_c \\ 0 & r \geq r_c \end{cases} \quad (2)$$

Recently, Fukuda and co-workers^{19,29} introduced a force-switching scheme for the Wolf summation in which electrostatic energy and force functions are switched to corresponding polynomials between a switching distance, r_1 , and the cutoff radius, r_c . In the force-switching Wolf (FSW) method, the interaction energy between two charges q_i and q_j in reduced units is given by

$$U_{\text{FSW}}(r) = \begin{cases} q_i q_j \left[\frac{\text{erfc}(\alpha r)}{r} + V^*(r_1) - \left[\frac{\text{erfc}(\alpha r_1)}{r_1} - \frac{V^*(r_1)}{2r_1} - \frac{\alpha}{\sqrt{\pi}} \right] q_i^2 \right] & r < r_1 \\ q_i q_j V^*(r) - \left[\frac{\text{erfc}(\alpha r_1)}{2r_1} - \frac{V^*(r_1)}{2r_1} + \frac{\alpha}{\sqrt{\pi}} \right] q_i^2 & r_1 \leq r \leq r_c \\ 0 & r > r_c \end{cases} \quad (3)$$

where

$$V^*(r) = p_1(r_c - r) + \frac{1}{2}p_2(r_c^2 - r^2) + \frac{1}{3}p_3(r_c^3 - r^3) + \frac{1}{4}p_4(r_c^4 - r^4) \quad (4)$$

and the pair force

$$F_{\text{FSW}}(r) = \begin{cases} q_i q_j \left[\frac{\text{erfc}(\alpha r)}{r^2} + \frac{2\alpha}{\sqrt{\pi}} \frac{\exp(-\alpha^2 r^2)}{r} \right] & r < r_1 \\ q_i q_j (p_1 + p_2 r + p_3 r^2 + p_4 r^3) & r_1 \leq r < r_c \\ 0 & r \geq r_c \end{cases} \quad (5)$$

where α is the damping factor and the polynomial coefficients p_i ($i = 1–4$), which are functions of α , r_1 , and r_c , are determined to satisfy the C^1 smoothness conditions at r_1 and r_c .²⁹ The force evaluated for $r_1 \leq r \leq r_c$ is a C^1 function at r_1 and smoothly decays to zero at the cutoff distance r_c .

Enhanced truncation methods have been tested on various systems. These include liquid water, NaCl crystals, and solutions;^{18,33} liquid–vapor water interfaces;^{23,26} globular proteins and membrane proteins immersed in lipid bilayers

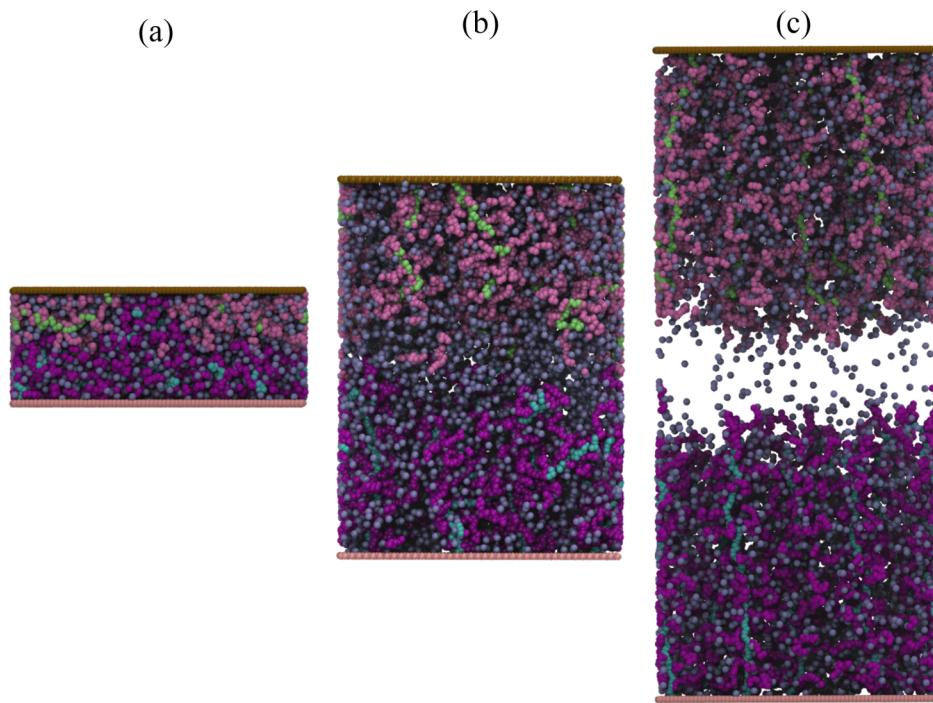


Figure 1. Snapshots of the system with grafting density $\rho_g = 3.09 \times 10^{-3} \sigma^{-2}$. The distance between the two substrates is (a) $D = 23\sigma$, (b) $D = 93\sigma$, and (c) $D = 163\sigma$. The system consists of 28 950 particles for the polymers and counterions and 5600 particles for each substrate. This gives a total of 40 150 particles.

with explicit water,²⁷ solvated fluorescent probes,²⁸ and zeolites.³⁴ These methods have also been employed as models in recent studies due to various advantages such as computational efficiency or the ability to simulate in a nonperiodic box.^{35–37} A comparison between the original Wolf method, the DSF method, and Ewald summation was recently performed by Hansen and co-workers,³³ showing that the DSF method provides not only accurate results but also a 2–3× performance speedup compared to the Wolf method. Yonezawa et al. employed the FSW method to study the free energy landscape of alanine dipeptide in explicit water.²⁹ They demonstrated that free energy landscapes of the peptide were highly comparable to those obtained by traditional Ewald methods.

In the present study, our objective is to evaluate the potential performance gains for enhanced truncation methods based on Wolf summation on hybrid high performance computers with GPU accelerators. Specifically, we describe our implementation of the DSF and FSW potentials into the LAMMPS molecular dynamics package³⁸ and compare performance with P³M for molecular dynamics simulation of polyelectrolyte brush layers on up to 512 Cray XK6 nodes with NVIDIA “Fermi” GPUs. We evaluate energetic, structural, and dynamic properties of systems of two charged bottle-brush polymer grafted surfaces based on recent work by the Dobrynin group.^{39–41} The polyelectrolyte brush coating surfaces have captured considerable interest during the past two decades^{42–46} due to their unique lubricating properties, reminiscent of biological connective tissues with fascinating low friction characteristics.⁴⁴ The nontrivial electrostatic interaction between the charged brushes with explicit counterions makes this system an excellent test case for the DSF method. Since the bottle-brush systems have many structural features in common with other systems with a slab geometry such as lipid bilayers in bulk water and ionic liquid interfaces, it is reasonable to expect that the findings herein could be relevant to the latter systems.

Additionally, we are able to evaluate configurations with different charge distribution profiles. It is known that truncating electrostatic interactions with small cutoffs for nonuniform charge distributions can lead to artifacts in the structural and dynamic properties,^{23,26,34,47–50} and therefore it is important to evaluate cases where the cutoff required for accurate simulation is too large to realize performance gains.

2. MODEL AND METHOD

2.1. Simulation Details. We employ the simulation model and method used in previous studies.^{39,40} In the coarse-grained model, polyelectrolyte bottle-brush macromolecules are represented by chains of charged Lennard-Jones particles with diameter σ . The bottle-brush macromolecule consists of a backbone chain with degree of polymerization $N_b = 97$ and 31 side chains each with 21 particles. The side chains are equally separated along the backbone. Only the side chains are charged with the fraction of charged monomers, $f = 1/3$, corresponding to every third monomer carrying a negative charge of $-1.0q^*$. $q^* = q/(4\pi\epsilon_0\epsilon_m\sigma k_B T)^{1/2}$ is the dimensionless charge, where ϵ_0 is the vacuum permittivity and ϵ_m is the dielectric constant of the solution. The macromolecules are tethered to two opposite substrates located at $z = 0$ and $z = D$, each of which is modeled by a periodic hexagonal lattice composed of 80×70 particles with diameter σ . The counterions are modeled as a spherical particle with a diameter of σ and a charge of $+1.0q^*$. The number of counterions is chosen so as to neutralize the whole system. Figure 1 shows example snapshots of the modeled system with 15 backbone chains grafting each 70σ -by- 80σ substrate, i.e., the grafting density of $\rho_g = 3.09 \times 10^{-3} \sigma^{-2}$.

The van der Waals interaction between all particles is modeled by the Lennard-Jones 12–6 potential, truncated, and shifted to zero at r_s . The cutoff $r_s = 2.5\sigma$ was chosen for polymer–polymer interactions and $r_s = 2^{1/6}\sigma$ for all other pairs.

The interaction strength ϵ is chosen to be $k_B T$ for polymer–counterion, polymer–substrate, counterion–counterion, and counterion–substrate interactions. For polymer–polymer interactions, we choose $\epsilon = 0.3k_B T$, which is close to a θ –solvent condition for the polymer backbone so as to minimize the effect of short-ranged attractions on the bottle-brush properties. The connectivity of monomers of the bottle-brush macromolecules and the attachment between the macromolecules and the substrate are modeled by the finite extension nonlinear elastic (FENE) potential.⁵¹

The electrostatic interaction between charged particles and counterions in a salt-free solution is modeled either using P³M as in the original studies,^{39,40} the DSF method as described in eq 1, or the FSW method as described in eq 3. Because P³M is implicitly periodic, artificial periodicity in the z dimension is removed by extending the P³M mesh with empty volume in the z dimension and removing dipole interactions between the x – y slabs.⁵² For the simulations here, the P³M mesh is 3 times as long as the simulation box in the z dimension. The system is equilibrated in the canonical ensemble using a Langevin thermostat with periodic boundary conditions in x and y directions. The parameters of the Langevin thermostat are identical with those in ref 39. The equation of motion of the particles is integrated by the velocity-Verlet algorithm with the time step of $\Delta t = 0.01\tau$. Here, $\tau = \sigma(m/\epsilon)^{1/2}$ is the reduced time unit and m is the particle mass.

2.2. DSF and FSW Implementation. We have implemented the DSF and FSW models into LAMMPS, an open-source massively parallel MD package.³⁸ The algorithms we have developed for efficient use of accelerators supporting CUDA or OpenCL in LAMMPS have been published in detail.^{30,53,54} LAMMPS supports acceleration for short-range force calculation⁵³ with optional acceleration for neighbor list builds and/or (P³M) long-range electrostatics.³⁰ Neighbor list builds are performed on the accelerator by first constructing a cell list that is utilized to build a Verlet list using a radix sort to assert deterministic results. The van der Waals and short-range electrostatic forces are computed in a separate kernel. For each particle, the force accumulation is performed by one or multiple threads. A default number of threads is chosen on the basis of the hardware and the potential model being used for calculation. For long-range electrostatics, acceleration for P³M is supported for charge assignment to the mesh and force interpolation. The parallel FFT is performed on the host. The P³M calculation can be performed at single or double precision.

All of the statistics computations, thermostats, barostats, time integration, bond/angle/dihedral/improper calculations, and any other simulation modifications are performed on the host. In order to achieve efficient acceleration, these calculations must be parallelized within each node on the host.⁵³ This is performed by using multiple MPI processes, each sharing one or more accelerators on a compute node. This approach has several advantages. Those relevant to the work here include full compatibility with all of the other routines in LAMMPS that run on the CPU, the ability to overlap data transfers and computation from different MPI processes sharing the accelerator, concurrent calculation of nonbonded forces on the accelerator and bonded forces on the host, and concurrent execution of long-range and short-range forces using separate partitions of MPI processes.⁵⁴ The downside of the approach is the requirement to determine an optimal number of MPI processes to share the accelerator. This will not necessarily be

all cores available on the node for smaller problem sizes due to the overhead for sharing an accelerator on current hardware.

Our implementation of the DSF and FSW potentials for LAMMPS is similar to that for other short-range models⁵³ with the exception of a few modifications required for efficient calculation of sparsely charged particles with relatively large cutoff distances. The first modification was support for an arbitrary cell size for neighbor list builds. In our previous work, the cell size was restricted to be the sum of the cutoff distance and skin specified for the model. When using a spatial decomposition with each cell assigned to a multiprocessor, this approach can minimize the number of computations for the Verlet list build and also the number of global memory fetches for particle positions when using cubic cells. With the large cutoff distances considered here, however, this can be detrimental to parallel performance because the number of cells for the neighbor list build can be much smaller than the number of available multiprocessors. The second modification was to support an arbitrary number of threads for force, energy, and virial computation for a given atom. In previous versions, this number was required to be less than or equal to the number of threads performing instructions in lock-step (for NVIDIA GPUs, this is known as the warp size). When using a number of threads that is greater than the warp size, additional synchronization barriers are required to assert that all threads have completed operations. However, it allows for an increase in the total number of threads performing the force calculation and thus is important for parallel performance. Finally, an option for ignoring uncharged particles that are beyond the short-range van der Waals force cutoff was implemented to allow for a single neighbor list for van der Waals and electrostatic force calculations on the accelerator.

2.3. DSF and FSW Evaluation. We evaluated the DSF and FSW models based on relative performance and accuracy when compared to P³M. For performance, we tested the model using the Titan development partition of the Jaguar supercomputer.⁵⁴ Jaguar is currently comprised of 18 688 Cray XK6 compute nodes and 512 service and I/O nodes. Each node uses a 16-core 64-bit AMD Opteron 6200 series processor with 32 GB of registered ECC DDR3 SDRAM. The nodes are connected in a 3D-torus by a Gemini interconnect with a submicrosecond latency for remote puts, 1–2 microsecond latency for other point-to-point messages, and 20 GB/s of injection bandwidth per node. Jaguar is scheduled to be upgraded to Titan, at which time nodes will also contain a total of 10–20 petaflops of NVIDIA “Kepler” GPU accelerators. Currently, however, the hybrid nodes are limited to the testing and development partition used here: 960 XK6 nodes with a 16-core Opteron 6200 series processor and a Tesla X2090 “Fermi” GPU. Performance was evaluated using the best run options for CPU-only P³M simulations, GPU-accelerated P³M simulations, and GPU-accelerated DSF/FSW simulations as determined by the runs described in the Results and Discussion section. All GPU-accelerated simulations used mixed precision calculations with double precision P³M as opposed to all double-precision for the CPU-only runs.

Accuracy for the DSF and FSW methods with respect to P³M was evaluated both by determining the error in the force calculation and by quantifying various structural and dynamic properties for bottle-brush simulations. For the former, we compute the difference in force magnitude and direction from the two methods for a given set of 100 independent configurations generated with P³M parametrized to have an

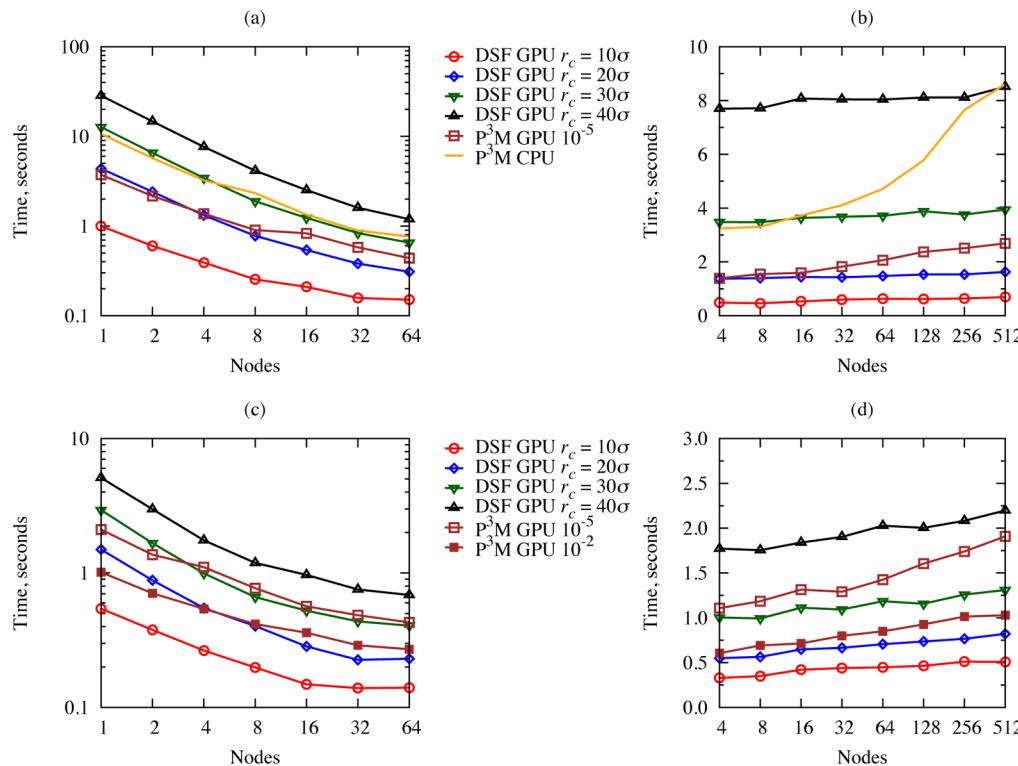


Figure 2. Benchmark results for the DSF and P³M methods on Cray XK6 nodes. Each node contains a 16-core AMD Opteron processor and a single Tesla X2090 GPU. (a) Strong scaling for 40 150 particles (cf. Figure 1b) and (b) weak scaling with 10 K particles per node using all particles for short-range calculations. (c,d) Strong and weak scaling results using only charged particles for the P³M direct summation. The grafting density is $\rho_g = 3.09 \times 10^{-3} \sigma^{-2}$.

estimated RMS error of less than $10^{-5} k_B T / \sigma$. In MD simulations, these two measures are highly relevant because they are representative of the underlying potential energy landscape that governs the system dynamic and equilibrium behaviors. The difference in force magnitudes is computed as a root mean squared error calculated using only the electrostatic contribution to the force for charged particles

$$(RMSE)^2 = \frac{1}{3N_c} \left\langle \sum_i^{N_c} (f_{i,x} - f_{i,x}^0)^2 + (f_{i,y} - f_{i,y}^0)^2 + (f_{i,z} - f_{i,z}^0)^2 \right\rangle \quad (6)$$

where $\langle \dots \rangle$ is the ensemble average, N_c is the number of charged particles, and f_i and f_i^0 are the electrostatic force components exerted on particle i obtained by the truncated-based methods (DSF or FSW) and P³M, respectively.

The difference in force directionality is characterized by the dot product between the force unit vectors given by the truncated-based methods (DSF or FSW) and by P³M. The alignment order parameter, S , is defined as¹⁸

$$S = \left\langle \frac{1}{N_c} \sum_i^{N_c} \hat{\mathbf{f}}_i \cdot \hat{\mathbf{f}}_i^0 \right\rangle = \left\langle \frac{1}{N_c} \sum_i^{N_c} \cos \gamma_i \right\rangle \quad (7)$$

where $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_i^0$ are the unit force vectors of the charged particle i evaluated by the truncated-based methods (DSF or FSW) and by P³M, respectively. The alignment order parameter, which is essentially the average of the cosine of the angle, γ , between two force unit vectors, indicates perfectly aligned force vectors when $S = 1$.

For structural and dynamic properties, we have followed the recent work by the Dobrynin group on bottle brushes. This included evaluation of the dependence of the shear velocity of the upper substrate on applied shear stress,⁴⁰ disjoining pressure between two polymer grafted substrates at various distances,³⁹ density profiles for counterions and charged monomers,³⁹ and various structural properties as described in the Results and Discussion.

3. RESULTS AND DISCUSSION

3.1. Performance. In order to assess the potential performance gains from using the DSF or FSW models, we performed parallel simulations of the bottle-brush systems using both P³M and the truncated models investigated here. For these benchmarks, the Titan development partition of the Jaguar supercomputer was used, with each node containing a 16-core AMD Opteron processor and a Tesla X2090 GPU. For “strong scaling” benchmarks, the size of the simulation is fixed at 40 150 particles, and the performance is evaluated using between 1 and 64 nodes. In the ideal case, the simulation time on 64 nodes is 1/64 of the time for simulation on one node. For “weak scaling” benchmarks, the performance is assessed at different system sizes. For these benchmarks, the 40 150 particle simulation is split between four nodes. Larger systems are obtained by replicating the simulation box in the x dimension x_r times and in the y dimension y_r times for a run on $4x_r y_r$ nodes. In the ideal case, the weak scaling simulation time for 5 139 200 particles on 512 nodes is identical to the simulation time for 40 150 particles on four nodes.

There are multiple parameters that can be adjusted that influence the performance of the simulations, and therefore many runs were performed to obtain the best parametrization for a given simulation. First, the cutoff length for the short-range electrostatics calculated with direct summation in real space can be varied. Increasing the cutoff allows for fewer mesh points to be used for the reciprocal space calculation in order to achieve the same accuracy. This shifts the computational work toward the short-range calculation and can decrease the communications bottleneck for parallel reciprocal space calculations. Second, LAMMPS now has an option to use separate partitions of MPI processes for the reciprocal space calculation.⁵⁴ This allows the reciprocal space calculation to be performed concurrently with the direct summation and also can decrease the number of MPI processes involved in communication for FFTs on each node. Third, the number of processes running on each node can be varied. For simulations using only the CPUs, it is typically more efficient to use all available cores. For GPU accelerated calculations, however, it can be more efficient to use fewer processes sharing the GPU on each node. Finally, the number of threads involved in force accumulation for each particle on the GPU can be varied.³⁰ The results presented here represent the best parametrization we found for each simulation case.

The results obtained when all particles are used in the short-range calculation are shown in Figure 2 for strong (a) and weak (b) scaling. Because the timings for FSW and DSF are indistinguishable on the plots, only DSF timings are shown. On a single node, the GPU accelerated P³M calculation is 2.86 times faster than the CPU-only calculation. Here, best results are obtained with 16 processes performing the CPU-only calculation with a cutoff of 10 σ . For the GPU-accelerated runs, the best performance is obtained with a cutoff of 18 σ using two processes for the reciprocal space calculations and an additional two processes for all other calculations. When using 16 or more nodes, the parallel efficiency of the GPU-accelerated runs suffers and the relative performance decreases. It also becomes more efficient to use a single process per node as the number of particles per node decreases.

In the weak scaling tests, the performance improvements of the P³M simulations with GPU acceleration become more significant for larger node counts, increasing from a speedup of 2.3 with four nodes to a speedup of 3.2 with 512 nodes. This result is somewhat counterintuitive but can be explained by the very high performance of GPU accelerators for the short-range direct summation. This allows for efficient calculations with relatively high cutoffs where the P³M mesh size is small and the reciprocal space communication bottleneck is reduced. At 512 nodes, we find it is most efficient to use a cutoff of 14 σ for CPU-only simulations. With GPU-acceleration, however, we can run efficiently with a cutoff of 24 σ allowing for a smaller number of mesh points.

For the DSF model, GPU-accelerated timings are shown for cutoffs of 10 σ , 20 σ , 30 σ , and 40 σ (Figure 2). In strong scaling tests, we found DSF with a cutoff of 10 σ to be between 2.9 and 3.7 times faster than GPU-accelerated P³M. With a cutoff of 20 σ , the results are similar to P³M with fewer than 16 nodes but significantly faster at higher node counts. For cutoffs of 20 σ and higher, the parallel efficiency for DSF simulations is significantly higher than for P³M. While at a cutoff of 10 σ , only 16 GPU threads per particle can be used efficiently, up to 64 threads per particle can be used for higher cutoffs, resulting in better GPU utilization for a fixed number of particles. For weak scaling

tests, parallel efficiency was better for DSF models in all cases when compared to P³M. At a cutoff of 10 σ , DSF simulations were between 2.7 and 3.8 times faster than GPU-accelerated P³M. At a cutoff of 20 σ , the DSF speedups ranged from 1.01 to 1.65.

The timings in Figure 2a,b are relevant for systems where all of the particles are charged. For the bottle-brush cases presented here, however, less than one-third of the particles have a charge, and the cutoff for van der Waals interactions is much smaller than for Coulombic interactions. Therefore, we also provide results in Figure 2c,d where only charged particles are included in the neighbor lists at distances larger than the van der Waals cutoff. Because there is not an exact C++ counterpart for these accelerator routines in LAMMPS, we do not include a P³M run without GPU acceleration. Instead, we show results for P³M parametrized to produce lower accuracy results. For the simulations in Figure 2a,b and all other P³M simulations in this paper, we have used the parametrization from previous studies that is estimated to result in an RMS error of 10⁻⁵ $k_B T/\sigma$. However, a smaller mesh can be used for P³M to achieve faster simulations with a less accurate force calculation. Therefore, we have also provided results with P³M parametrized to an estimated RMS error of less than 10⁻² $k_B T/\sigma$.

Because less time is spent in the short-range direct summation when only the charged particles are included, the reciprocal space calculation represents a larger fraction of the simulation time, and therefore the speedups resulting from use of the DSF model are more significant as shown in Figure 2c,d. In this case, cutoffs as large as 30 σ can be faster than the high accuracy P³M. For a cutoff of 10 σ , DSF was 3.1 to 3.9 times faster than P³M in strong scaling tests and 3.4 to 3.7 times faster for weak scaling. For 20 σ , the benchmarks showed speedups ranging from 1.4 to 2.32 for DSF. Decreasing the P³M accuracy to an RMS error of 10⁻² $k_B T/\sigma$ can result in significantly faster simulations; however, they are still slower than DSF with a cutoff of 20 σ in most cases. For a cutoff of 10 σ , the strong scaling benchmarks resulted in speedups of 1.87 to 1.93. At 20 σ , DSF was up to 18% faster when using greater than four nodes and slower otherwise. For weak scaling, speedups were 1.8 to 2 for 10 σ DSF and 1.1 to 1.3 for 20 σ DSF. Because there are few mesh points per process with the low accuracy P³M parametrization used here, increasing the short-range cutoff beyond 10 σ offers little performance gain when compared to the high accuracy P³M simulations, and there is little performance to be gained with further decreases in P³M accuracy based on the current LAMMPS implementation.

3.2. Shear Velocity versus Shear Stress. The sliding velocity is central for calculation of the friction between two brushed surfaces because it is directly related to the shear rate, and from that, to shear viscosity. Studies on the dependence of the shear velocity of the upper substrate on the applied shear stress with electrostatic interactions evaluated by P³M have been reported previously.⁴⁰ Here, we assess the ability of the DSF and FSW models to reproduce these results. In these simulations, the particles in the upper substrate are subject to two constant forces: one parallel to the lower substrate, f_y , and the other perpendicular to the substrate along the negative z direction, f_z . The shear and compression stresses are defined as $\sigma_{yz} = f_y/a_0$ and $\sigma_{zz} = f_z/a_0$, respectively, where $a_0 = 0.866\sigma^2$ is the area per particle forming the substrates. The shear velocity is determined from the displacement of the center of mass of

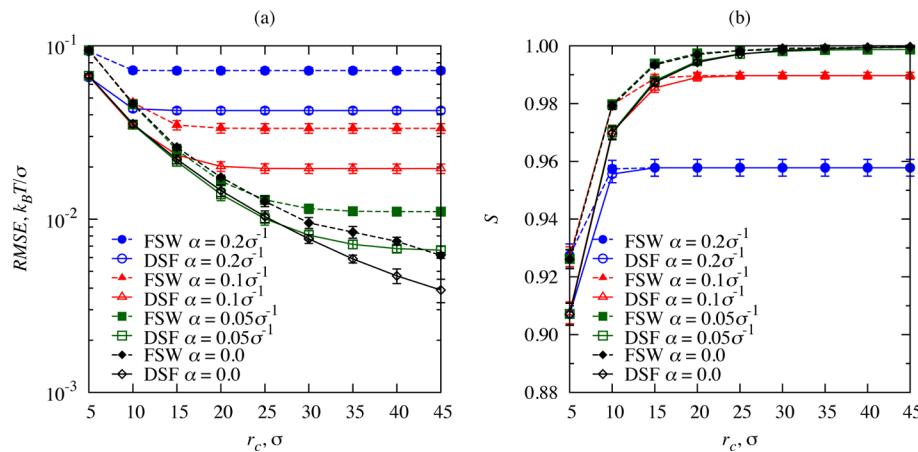


Figure 3. Accuracy in force magnitude (a) and direction (b) obtained from the DSF and FSW models using P^3M as the reference. Accuracy is plotted as a function of the cutoff radius, r_c , and damping factor, α , for $\rho_g = 1.24 \times 10^{-3} \sigma^{-2}$. Configurations are taken from simulations with a shear stress of $\sigma_{yz} = 0.00288 k_B T / \sigma^3$.

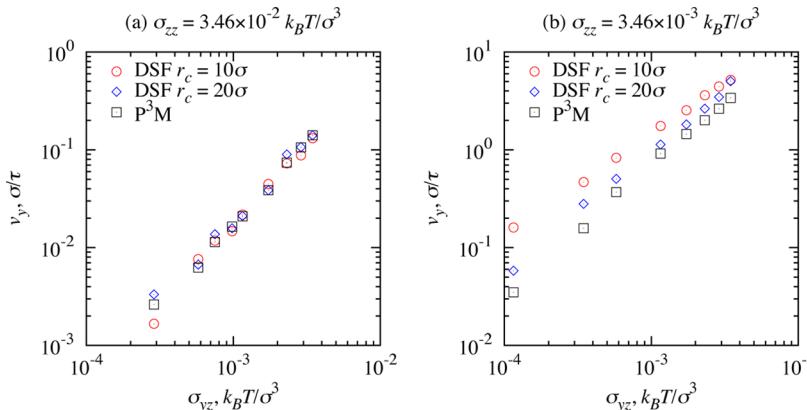


Figure 4. Dependence of the shear velocity of the upper substrate versus shear stress, σ_{yz} , for grafting density $\rho_g = 1.24 \times 10^{-3} \sigma^{-2}$: (a) at the high compression stress and (b) at the low compression stress. The damping factor in the DSF model is chosen as $\alpha = 0.05 \sigma^{-1}$.

the upper substrate along the y direction at the steady state regime.

In order to assess the accuracy of the DSF and FSW models in terms of force magnitude and directionality, we consider a set of 50 configurations that have been previously equilibrated. We evaluate the forces on each charged particle using the DSF model, the FSW model, and P^3M . Contributions from van der Waals and bonded interactions are not included in the error calculation. Using this approach, we obtain the results in Figure 3 for a polymer grafting density of $\rho_g = 1.24 \times 10^{-3} \sigma^{-2}$. For both force magnitude and direction, a comparison of the DSF and FSW models reveals better accuracy with the DSF model for bottle-brush systems for all of the cases we investigated. Therefore, we have restricted all other analysis in the paper to the DSF model for clarity. Figure 3a shows accuracy for different cutoffs and damping factors. Of course, the accuracy increases with cutoff distance. Additionally, we find that the optimal value for the damping factor decreases with an increasingly large cutoff. Using $r_c = 10\sigma$ leads to an RMS error as low as $0.035 k_B T / \sigma$, i.e. 3.5% of the force magnitude due to thermal fluctuations. Because thermal energy sets the force scale in this system, this RMS error could be considered acceptable. Figure 3b shows the accuracy in force direction. As expected, results also improve upon decreasing α and increasing r_c . At $r_c = 15\sigma$ and $\alpha = 0.05 \sigma^{-1}$, we find that $S = 0.989 \pm 0.001$,

or equivalently, the angle between the force vector calculated by the DSF method and that by P^3M is approximately 8.5° .

Figure 4a demonstrates that the dependence of shear velocity on shear stress can be reproduced by the DSF method with high fidelity. The shear velocity curves obtained with $r_c = 10\sigma$ and 20σ are almost indistinguishable with that from the P^3M method. The results in Figure 3 and Figure 4a are taken from shear simulations with a high compression stress. If we examine results using a low compression stress (Figure 4b), a significant difference can be seen between the models. Using a cutoff as large as 20σ only yields consistent results with P^3M at sufficiently high shear stress, i.e. $\sigma_{yz} > 10^{-3} k_B T / \sigma^3$. Consequently, for the low compression stress the P^3M method should be used rather than the DSF and FSW methods with this cutoff. We note that at the steady state, the separation between two substrates at high compression stress is $D \approx 20\sigma$ for either $r_c = 10\sigma$ or $r_c = 20\sigma$ for the whole range of σ_{yz} under consideration. However, at the low compression stress, the steady-state distance between two substrates under the same shear stress varies with r_c noticeably: for instance, at $\sigma_{yz} = 0.0025 k_B T / \sigma^3$, $D = (163 \pm 6.8)\sigma$ with $r_c = 10\sigma$, whereas $D = (100 \pm 3.9)\sigma$ with $r_c = 20\sigma$. Because the increased distance between the substrates weakens the electrostatic interaction between them, an increase in the sliding velocity is therefore understandable. At low compression stress, the distance between the substrate planes is significantly higher allowing

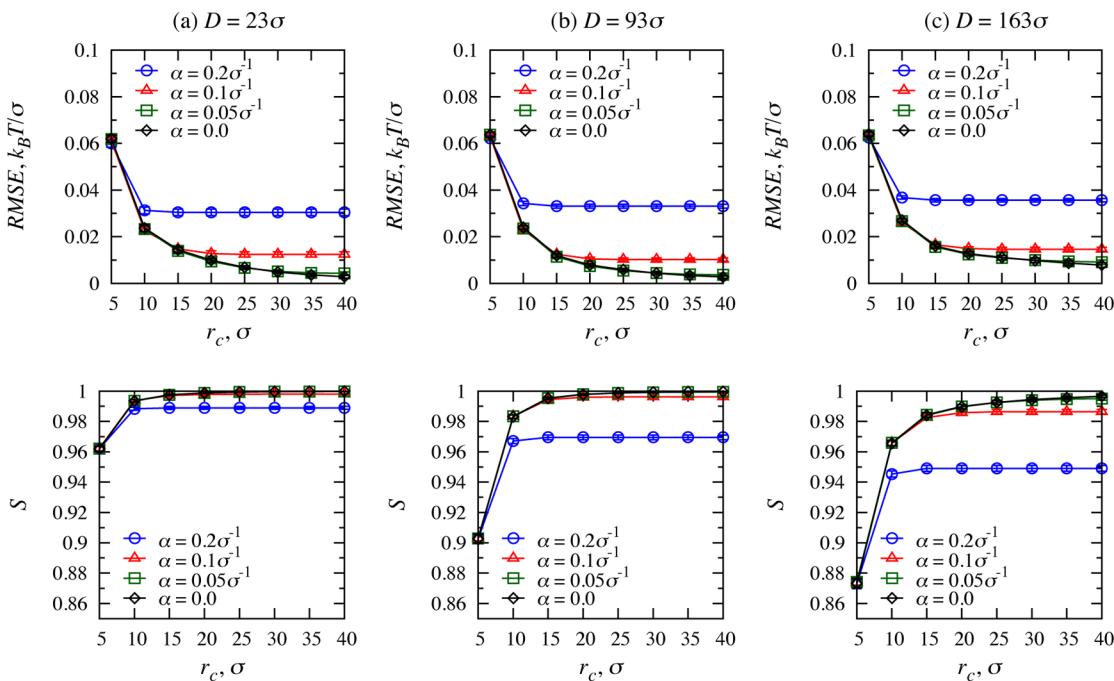


Figure 5. Accuracy in force magnitude (top) and direction (bottom) obtained from the DSF method compared with P^3M as functions of the cutoff radius, r_c , and damping factor, α , for $\rho_g = 3.09 \times 10^{-3} \sigma^{-2}$ at different separations between two substrates.

for more complex geometries of charged particles. Because these geometries can impact model accuracy, we continue our analysis by investigating the disjoining pressure between the two substrates as a function of separation.

3.3. Disjoining Pressure versus Separation. The disjoining pressure between two surfaces grafted by charged polyelectrolyte brushes was shown to build up even when the brushes had not been in physical contact with each other.³⁹ The extremely low friction between two surfaces was attributed to the free counterion layer which prevented the interpenetration between brushes.⁴⁴ For these simulations to evaluate disjoining pressure with the DSF model, we fixed the distance between two substrates and measured the time-averaged force exerted on the substrate particles along the z direction, $\langle f_z \rangle$. The disjoining pressure is defined as $P = \langle f_z \rangle / a_0$, where $a_0 = 0.866\sigma^2$ is the area per substrate bead as described earlier. The forces are averaged over 10^4 samples during simulation runs lasting $10^4\tau$.

The force magnitude and direction obtained by the DSF method are first compared against those by the P^3M method at three different values of substrate separation, D . For this analysis, we consider a set of 100 configurations equilibrated by the P^3M method at each separation and switch to the DSF method for the calculation of the electrostatic forces. The Lennard-Jones and bond forces are not included in the error calculation. The grafting density of the polymers for this analysis is $\rho_g = 3.09 \times 10^{-3} \sigma^{-2}$.

The accuracy in force magnitude and direction as functions of r_c and α is shown in Figure 5. For small separations, $D = 23\sigma$ and $D = 93\sigma$, the DSF method gives fairly accurate forces as compared with those obtained by the P^3M method. For $D = 23\sigma$, with $r_c = 10\sigma$ the RMSE is as low as $0.023k_B T/\sigma$, i.e. 2.3% of the force magnitude due to thermal fluctuations. The accuracy in force directionality increases upon decreasing α and increasing r_c . For instance, for $D = 23\sigma$, using $r_c = 5\sigma$ and $\alpha = 0.05\sigma^{-1}$, $S = 0.962 \pm 0.001$, or equivalently, the angle between

the force vector calculated by the DSF method and that by P^3M is approximately 15.8° . The difference is reduced to 6.5° when using $r_c = 10\sigma$.

For the large separation, $D = 163\sigma$, the calculated electrostatic forces for the DSF model become less accurate (Figure 5c, bottom) at a given cutoff. Using $r_c = 10\sigma$ and $\alpha = 0.05\sigma^{-1}$, the error in the force magnitude is $0.027k_B T/\sigma$, yet the error in the force direction is $S = 0.966 \pm 0.001$, or equivalently, 15° . To reduce the force direction error to 6.5° , the cutoff radius should be as large as 30σ . However, at this cutoff, P^3M parametrized to similar accuracy is faster on the hardware we used for the study (Figure 2).

At large distances, the bottle-brush layers become separated allowing for a layer of positively charged counterions to form.³⁹ In these cases, Coulombic screening can be diminished and therefore truncating the electrostatics with small cutoffs can lead to significant errors. This is shown in Figure 6, where the disjoining pressure is calculated for a range of distances. For the small separation regime where the bottle-brush layers overlap,

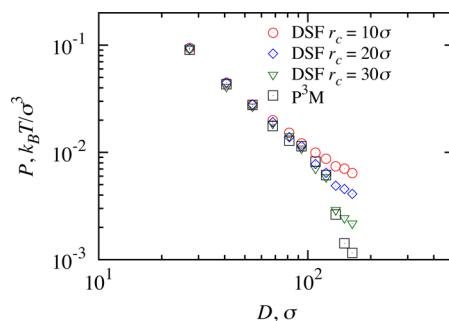


Figure 6. Disjoining pressure versus distance between two substrates for $\rho_g = 3.09 \times 10^{-3} \sigma^{-2}$ using P^3M or DSF with different cutoff radii. The damping factor in the DSF model is chosen as $\alpha = 0.05\sigma^{-1}$. Error bars are excluded for clarity.

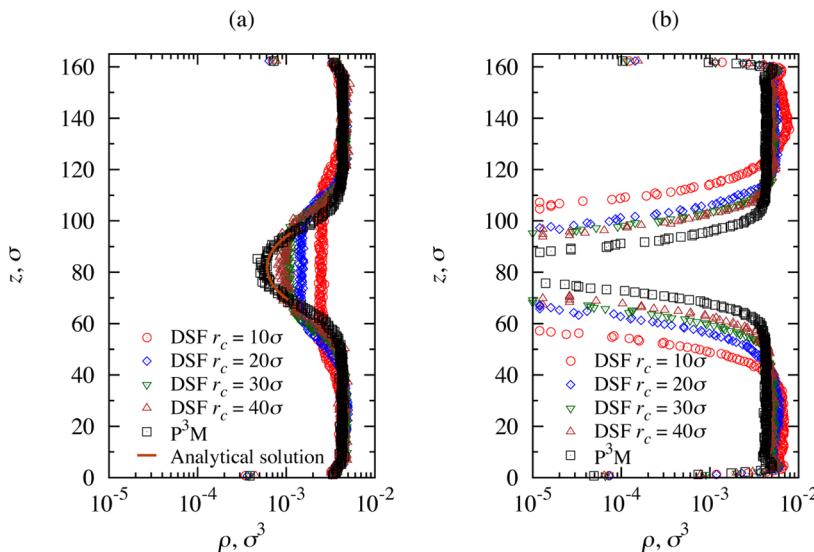


Figure 7. Density profiles of (a) counterions and (b) charged monomers for $D = 163\sigma$ at the grafting density of $\rho_g = 1.24 \times 10^{-3}\sigma^{-2}$.

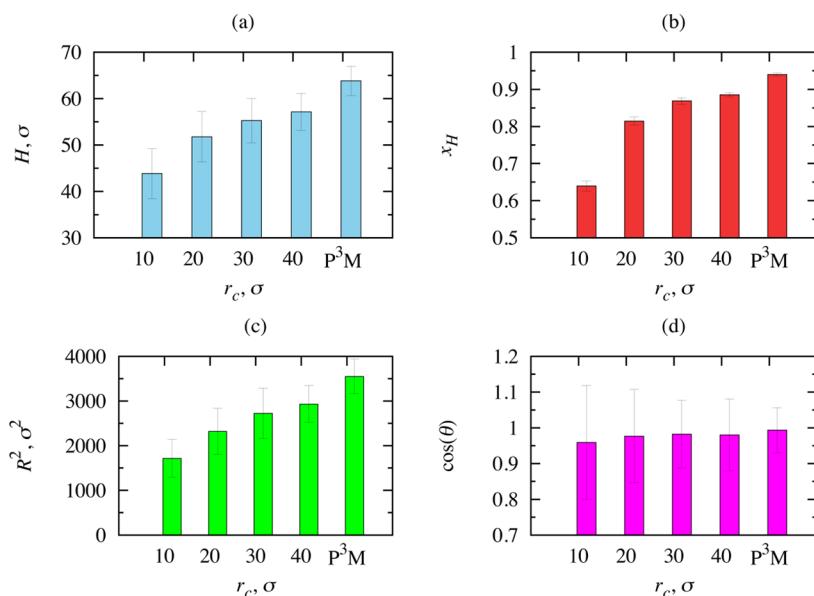


Figure 8. Structural properties for different cutoff radii obtained by the DSF method compared with those by P³M: (a) brush height, (b) counterion fraction in the brushes, (c) squared end-to-end distance of the backbone, and (d) cosine of the tilted angle of the backbone chains. $\rho_g = 1.24 \times 10^{-3}\sigma^{-2}$ and $D = 163\sigma$. Error bars are obtained by averaging over 500 samples with 5.0τ intervals.

$D \leq 100\sigma$, the disjoining pressure computed from the DSF method matches with that from the P³M method considerably well. Figure 6a demonstrates the consistency between two methods for $\rho_g = 3.09 \times 10^{-3}\sigma^{-2}$. The pressure data points at each D were obtained from relaxing the P³M-equilibrated structures for $2 \times 10^4\tau$, or equivalently, 2×10^6 time steps, using the DSF method with the corresponding r_c . In the large separation regime ($D \geq 100\sigma$), the DSF model in general predicts higher values of the disjoining pressure (Figure 6b). As the cutoff radius is increased, the pressure values converge to those obtained from P³M. However, at the large separations, it is more efficient to use P³M.

3.4. Structural Properties. We can quantify the irregularity in the charge distribution for the bottle-brush systems at large separations by calculating the density profiles of the counterions and of the charged monomers. This is shown in Figure 7 for the largest separation, $D = 163\sigma$. The

profiles are obtained by averaging the 100 samples equilibrated by the DSF method with an interval of 100τ . For comparison, the initial P³M equilibrated profiles are plotted in black squares.

For small r_c , the DSF method yields a higher and uniform counterion density within the middle layer (Figure 7a); meanwhile, the profile predicted by the P³M method is highly consistent with the analytical solution.^{39,55} The brush layers are compressed toward the substrate, as indicated by the density profile of the charged monomers (Figure 7b). When r_c is increased up to 40σ , the density profiles asymptotically recover those obtained by P³M. Note that a cutoff of $r_c = 40\sigma$ covers half of the substrate dimensions of $70\sigma \times 80\sigma$, and slightly greater than the thickness of the interfacial counterion layer predicted by the P³M method, $\approx 35\sigma$.

Truncating the electrostatic potentials at large separations influences not only the disjoining pressure due to electrostatic forces along the z dimension but also can lead to artifacts in the

structural properties of the system. Figure 8 illustrates this problem at the largest separation for several important structural properties using the DSF method with different cutoffs. Consistent with the density profiles, the height of the brush layers, H , and the fraction of counterions in the brushes, x_H , both decrease with r_c . For instance, for $r_c = 10\sigma$, the height of the brush is $H = (43.8 \pm 5.3)\sigma$, remarkably smaller than predicted by P³M, i.e. $(63.8 \pm 3.1)\sigma$. Increasing r_c up to 40σ yields a brush layer height within the statistical error of the P³M value. The same trend is observed for the end-to-end distance and tilt angle of the backbone chains. The reduced fraction of counterions in the brush layers (Figure 8b) and their increased density in the midplane region (Figure 7a) indicate that the counterions were withdrawn from the brushes, leaving the brushes non-neutralized. These results clearly demonstrate that the DSF model, without a sufficiently large cutoff, causes artifacts at large D including the more bent and tilted backbone chains and the escape of counterions into the midplane region. As pointed out by Russano et al.,³⁹ the bending energy of the backbone chains contributes a large part to the disjoining pressure and therefore can contribute to the deviations shown in Figure 6.

4. CONCLUSION

Due to their simplicity, enhanced truncation approaches for evaluation of electrostatic interactions in condensed-phase systems are a tempting consideration as alternative methods that can exploit the massive concurrency available on modern computing architectures. Indeed, with relatively minor modifications to the LAMMPS molecular dynamics code, we were able to achieve considerable performance improvements with the DSF and FSW models using GPU accelerators. The advantages of these types of approaches are expected to become more pronounced on future hardware. Already, there are six times as many cores per multiprocessor on current NVIDIA GPUs when compared to the test hardware used for this study. The same trend holds on supercomputers without GPU accelerators; the current record holder for peak performance has 1 572 864 cores.⁵⁶

The use of truncated methods is justified by the screening of electrostatic interactions in condensed-phase systems. In this regard, we have chosen a particularly challenging system for evaluation of the DSF model—coarse-grained representation of charged bottle-brush covered surfaces. Nonetheless, we were able to demonstrate that for many of the simulations we could obtain accurate results using cutoffs that result in faster run times than the simulations using P³M. The errors in force magnitude and direction with the DSF method are as small as 2–3% of thermal fluctuations and 3–6°, respectively, using a practical cutoff, i.e., $r_c = 15\text{--}20\sigma$.

We can summarize our key findings as follows. (1) Because GPU accelerators allow for efficient calculation with much larger cutoffs, parallel efficiency for simulations with P³M can be drastically improved with GPU-acceleration owing to the smaller mesh size. (2) With GPU acceleration, the FSW and DSF models allow for better parallel efficiency than simulations with P³M up to relatively large cutoffs. This includes better parallel performance with fewer particles per node and better parallel performance for larger systems run on more nodes. (3) Convergence for the DSF model was better than for the FSW model in all cases for the bottle-brush systems used here. (4) At low separations, simulation of bottle-brush bilayers with DSF can produce accurate results as measured by force magnitudes

and directions, shear velocities, and disjoining pressures with cutoffs that result in significantly faster simulation times when compared to P³M. (5) At large separations, nonuniform charge distributions due to the formation of counterion layers require DSF cutoffs that are too large for efficient simulation. In these cases, the use of smaller cutoffs results in artifacts in brush morphology and counterion distribution and produces inaccurate results.

Although the potential advantages of truncated models such as DSF seem clear, validation of the model has been limited and mostly empirical. Although these studies are useful to evaluate limitations of the model on representative systems, it is clearly desirable to develop a formalism to better quantify the concept of screening in condensed-phase systems or at the least, a metric to detect when truncated models result in dangerous approximations. For the latter, examples include comparison to P³M at some time step interval that still results in efficient simulation or perhaps spectral analysis of charge within the system. Research along this direction could also be advantageous in further improving performance with the development of approaches capable of balancing work with more computation in regions of the simulation box sensitive to errors.

In order to help facilitate further research in this area, we have included the DSF model used in this paper within the LAMMPS software package. We again emphasize, however, that the model should not be used without appropriate validation; at the time of writing, methods to detect when the parametrization of the model might give rise to erroneous results were not in place.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: brownw@ornl.gov.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

All of the code described in this paper is available in the open-source LAMMPS software package, available at <http://lammps.sandia.gov/> or by contacting the authors. This research was conducted in part under the auspices of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC. This research used resources of the Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

■ REFERENCES

- (1) Frenkel, D.; Smit, B. *Understanding Molecular Simulations: Theory and Applications*, 2nd ed.; Academic Press: London, 2002.
- (2) Darden, T.; York, D.; Pedersen, L. *J. Chem. Phys.* **1993**, *98*, 10089.
- (3) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. *J. Chem. Phys.* **1995**, *103*, 8577–8593.
- (4) Hockney, R. W.; Eastwood, J. W. *Computer Simulation Using Particles*; IOP: Bristol, United Kingdom, 1988.
- (5) Schulz, R.; Lindner, B.; Petridis, L.; Smith, J. C. *J. Chem. Theory Comput.* **2009**, *5*, 2798–2808.
- (6) Cerutti, D. S.; Duke, R. E.; Darden, T. A.; Lybrand, T. P. *J. Chem. Theory Comput.* **2009**, *5*, 2322–2338.

- (7) Harvey, M. J.; Fabritiis, G. D. *J. Chem. Theory Comput.* **2009**, *5*, 2371–2377.
- (8) Shan, Y.; Klepeis, J. L.; Eastwood, M. P.; Dror, R. O.; Shaw, D. E. *J. Chem. Phys.* **2005**, *122*, 054101.
- (9) Hardy, D. J.; Stone, J. E.; Schulten, K. *Parallel Comput.* **2009**, *35*, 164–177.
- (10) Neumann, M. *J. Chem. Phys.* **1985**, *82*, 5663–5672.
- (11) Gnanakaran, S.; Nussinov, R.; Garcia, A. E. *J. Am. Chem. Soc.* **2006**, *128*, 2158–2159.
- (12) Matthes, D.; de Groot, B. L. *Biophys. J.* **2009**, *97*, 599–608.
- (13) Weeks, J. D.; Selinger, R. L. B.; Broughton, J. Q. *Phys. Rev. Lett.* **1995**, *75*, 2694–2697.
- (14) Weeks, J. D.; Katsov, K.; Vollmayr, K. *Phys. Rev. Lett.* **1998**, *81*, 4400–4403.
- (15) Wu, X.; Brooks, B. R. *J. Chem. Phys.* **2005**, *122*, 044107.
- (16) Zahn, D.; Schilling, B.; Kast, S. M. *J. Phys. Chem. B* **2002**, *106*, 10725–10732.
- (17) Wolf, D.; Kebinski, P.; Phillipot, S. R.; Eggebrecht, J. *J. Chem. Phys.* **1999**, *110*, 8254–8282.
- (18) Fennell, C. J.; Gezelter, J. D. *J. Chem. Phys.* **2006**, *124*, 234104.
- (19) Fukuda, I.; Yonezawa, Y.; Nakamura, H. *J. Phys. Soc. Jpn.* **2008**, *77*, 114301.
- (20) Wu, X.; Brooks, B. R. *J. Chem. Phys.* **2008**, *129*, 154115.
- (21) Rodgers, J. M.; Weeks, J. D. *Proc. Natl. Acad. Sci. U.S.A.* **2008**, *105*, 19136–19141.
- (22) Wu, X.; Brooks, B. R. *J. Chem. Phys.* **2009**, *131*, 024107.
- (23) Takahashi, K. Z.; Narumi, T.; Yasuoka, K. *J. Chem. Phys.* **2011**, *134*, 174112.
- (24) Yonezawa, Y. *J. Chem. Phys.* **2012**, *136*, 244103.
- (25) Orsi, M.; Haubertin, D. Y.; Sanderson, W. E.; Essex, J. W. *J. Phys. Chem. B* **2008**, *112*, 802–815.
- (26) Mendoza, F. N.; López-Lemus, J.; Chapela, G. A.; Alejandre, J. J. *Chem. Phys.* **2008**, *129*, 024706.
- (27) Kikugawa, G.; Apostolov, R.; Kamiya, N.; Taiji, M.; Himeno, R.; Nakamura, H.; Yonezawa, Y. *J. Comput. Chem.* **2009**, *30*, 110–118.
- (28) Furse, K. E.; Corcelli, S. A. *J. Chem. Theory Comput.* **2009**, *5*, 1959–1967.
- (29) Yonezawa, Y.; Fukuda, I.; Kamiya, N.; Shimoyama, H.; Nakamura, H. *J. Chem. Theory Comput.* **2011**, *7*, 1484–149.
- (30) Brown, W. M.; Kohlmeyer, A.; Plimpton, S. J.; Tharrington, A. N. *Comput. Phys. Commun.* **2012**, *183*, 449–459.
- (31) Demontis, P.; Spanu, S.; Suffritti, G. B. *J. Chem. Phys.* **2001**, *114*, 7980–7988.
- (32) Angoshtari, A.; Yavari, A. *Phys. Lett. A* **2011**, *375*, 1281–1285.
- (33) Hansen, J. S.; Schroder, T. B.; Dyre, J. C. *J. Phys. Chem. B* **2012**, *116*, 5738–5743.
- (34) Vlugt, T. J. H.; García-Pérez, E.; Dubbeldam, D.; Ban, S.; Calero, S. *J. Chem. Theory Comput.* **2008**, *4*, 1107–1118.
- (35) Orsi, M.; Essex, J. W. *PLoS ONE* **2011**, *6*, e28637.
- (36) Bresme, F.; Lervik, A.; Bedeaux, D.; Kjelstrup, S. *Phys. Rev. Lett.* **2008**, *101*, 020602.
- (37) Nagata, Y.; Mukamel, S. *J. Am. Chem. Soc.* **2010**, *132*, 6434–6442.
- (38) Plimpton, S. *J. J. Comput. Phys.* **1995**, *117*, 1–19.
- (39) Russano, D.; Carrillo, J.-M. Y.; Dobrynin, A. V. *Langmuir* **2011**, *26*, 18374–18381.
- (40) Carrillo, J.-M. Y.; Russano, D.; Dobrynin, A. V. *Langmuir* **2011**, *27*, 14599–14608.
- (41) Ou, Y.; Sokoloff, J. B.; Stevens, M. *J. Phys. Rev. E* **2012**, *85*, 011801.
- (42) Raviv, U.; Giasson, S.; Kampf, N.; Gohy, J. F.; Jerome, R.; Klein, J. *Nature* **2003**, *425*, 163–165.
- (43) Ballauff, M.; Borisov, O. *Curr. Opin. Colloid Interface Sci.* **2006**, *11*, 316–323.
- (44) Klein, J. *Proc. IME Part J: J. Eng. Tri.* **2006**, *220*, 691–710.
- (45) Lee, S.; Spencer, N. D. *Science* **2008**, *319*, 575–576.
- (46) Chen, M.; Briscoe, W. H.; Armes, S. P.; Klein, J. *Science* **2009**, *323*, 1698–1700.
- (47) Loncharich, R. J.; Brooks, B. R. *Proteins: Struct., Funct., Genet.* **1989**, *6*, 32–45.
- (48) Patra, M.; Karttunen, M.; Hyvonen, M. T.; Falck, E.; Lindqvist, P.; Vattulainen, I. *Biophys. J.* **2003**, *84*, 3636–3645.
- (49) Bergdorf, M.; Peter, C.; Hunenberger, P. H. *J. Chem. Phys.* **2003**, *119*, 9129–9144.
- (50) Wong-Ekkabut, J.; Karttunen, M. *J. Chem. Theory Comput.* **2012**, *8*, 2905–2911.
- (51) Kremer, K.; Grest, G. S. *J. Chem. Phys.* **1990**, *92*, 5057–5086.
- (52) Yeh, I.-C.; Berkowitz, M. L. *J. Chem. Phys.* **1999**, *111*, 3155–3162.
- (53) Brown, W. M.; Wang, P.; Plimpton, S. J.; Tharrington, A. N. *Comput. Phys. Commun.* **2011**, *182*, 898–911.
- (54) Brown, W. M.; Nguyen, T. D.; Fuentes-Cabrera, M.; Fowlkes, J. D.; Rack, P. D.; Berger, M.; Bland, A. S. *Procedia Comput. Sci.* **2012**, *9*, 186–195.
- (55) Safran, S. A. *Statistical Thermodynamics of Surfaces, Interfaces, and Membranes*; Westview Press: Boulder, CO, 2003.
- (56) Top500 Supercomputer Sites. <http://www.top500.org>. (accessed October 9, 2012).