Published in final edited form as:

J Comput Chem. 2011 November 15; 32(14): 2958–2973. doi:10.1002/jcc.21871.

Structural, Dynamic, and Electrostatic Properties of Fully Hydrated DMPC Bilayers From Molecular Dynamics Simulations Accelerated with Graphical Processing Units (GPUs)

Narayan Ganesan¹, Brad A. Bauer², Timothy R. Lucas², Sandeep Patel^{2,*}, and Michela Taufer^{1,*}

¹Department of Computer and Information Science, University of Delaware, Newark, Delaware 19716, USA

²Department of Chemistry and Biochemistry, University of Delaware, Newark, Delaware 19716, USA

Abstract

We present results of molecular dynamics simulations of fully hydrated DMPC bilayers performed on graphics processing units (GPUs) using current state-of-the-art non-polarizable force fields and a local GPU-enabled molecular dynamics code named FEN ZI. We treat the conditionally convergent electrostatic interaction energy exactly using the Particle Mesh Ewald method (PME) for solution of Poisson's Equation for the electrostatic potential under periodic boundary conditions. We discuss elements of our implementation of the PME algorithm on GPUs as well as pertinent performance issues. We proceed to show results of simulations of extended lipid bilayer systems using our program, FEN ZI. We performed simulations of DMPC bilayer systems consisting of 17004, 68484 and 273936 atoms in explicit solvent. We present bilayer structural properties (atomic number densities, electron density profiles), deuterium order parameters (S_{CD}), electrostatic properties (dipole potential, water dipole moments), and orientational properties of water. Predicted properties demonstrate excellent agreement with experiment and previous all-atom molecular dynamics simulations. We observe no statistically significant differences in calculated structural or electrostatic properties for different system sizes, suggesting the small bilayer simulations (less than 100 lipid molecules) provide equivalent representation of structural and electrostatic properties associated with significantly larger systems (over 1000 lipid molecules). We stress that the three system size representations will have differences in other properties such as surface capillary wave dynamics or surface tension related effects that are not probed in the current study. The latter properties are inherently dependent on system size. This contribution suggests the suitability of applying emerging GPU technologies to studies of an important class of biological environments, that of lipid bilayers and their associated integral membrane proteins. We envision that this technology will push the boundaries of fully atomic-resolution modeling of these biological systems, thus enabling unprecedented exploration of meso-scale phenomena (mechanisms, kinetics, energetics) with atomic detail at commodity hardware prices.

I. INTRODUCTION

Integral membrane proteins have become an arena of intense effort of fundamental research due to the ubiquitous nature of these macromolecules. Comprising roughly a third of the human genome, they are implicated in myriad physiological function and, unfortunately,

*Corresponding author. sapatel@udel.edu, taufer@udel.edu.

dysfunction. For instance, for normal physiological functioning, integral membrane proteins are involved in signaling processes, passive and active transport, and interfacial enzymatic processes¹. Of course, one cannot speak about integral membrane proteins independent of the lipidic context in which they function. Lipid membranes in their own right have garnered much attention as well, with particular focus on membrane properties and behaviors including structural deformation (in the presence of small molecules as well as integral membrane proteins), and electrostatic properties such as the interfacial dipole (or total) potential² and dielectric constant variation with location in a bilayer³. Of particular recent interest have been protein-lipid interactions, and specifically, the interactions of charged and/or polar amino acid residues as they pertain to an understanding of the thermodynamics of structural and energetic stability of integral membrane proteins upon desolvation of such systems.

Complementing the enormous volume of experimental effort to understand lipid bilayers and integral membrane proteins, computational approaches based on the analytics of statistical mechanics (molecular dynamics and Monte Carlo simulations) have been an indispensable tool in understanding properties and processes in these systems at the atomic and molecular level, thus providing insights at a resolution inaccessible to current state of the art experimental methods^{1,4-16}. These methods require information about the forces between interacting species; ideally, the necessary information would be contained in a quantum mechanical potential energy surface, which could then be directly applied to generate forces. However, simplifying assumptions must be made in order to arrive at tractable functional forms of a classical nature. Current state-of-the-art modeling approaches, thus, employ empirical potentials, or force fields, to model inter-species interactions. Though initially developed with a fully atomistic perspective, with secondary efforts pursuing more “coarse-grained” potentials^{17,18}, development and application of coarse-grained models continues at a rigorous pace today¹⁹⁻²⁴. Nevertheless, under the current paradigmatic approach to force field development, such models are generally referenced to results from fully atomistic simulations. Thus, the need for accurate all-atom potentials still persists. Moreover, the use of fully atomistic descriptions of model systems at meso-scales (here taken to be intermediate between the atomic and/or molecular scales of tens to hundreds of Angstroms and continuum descriptions) would enable scientific inquiry into atomic-level mechanisms and processes for which coarse-grained approaches would offer too low a resolution. Questions regarding the association of several peptides in bilayers, lipid damage, roles of meso-scale domains (‘rafts’) in signaling and bilayer structure and function are just some of the types of questions that would benefit from the ability to routinely simulate on the meso-scale with atomistic detail.

While fully-atomistic models are continually being refined to greater accuracy, they are still limited by traditional computer platforms that, because of their resource constraints, ultimately impact the models’ level of realism and the computational (time- and length-) scales achievable. With the advent of multi-core platforms such as graphics processing units (GPUs), scientists now have new computing tools to overcome these barriers²⁵⁻³⁸. At the same time the migration of these biological models to the new platforms include new challenges, including redesigning and implementing computational algorithms to meet the new platform configurations as well as assessing and validating the simulation results. This paper addresses both challenges by presenting the algorithmic implementation of an efficient code for Molecular Dynamics (MD) simulations for GPUs including Particle Mesh Ewald (PME)^{39,40} and presenting a detailed analysis of the energy profiles for DMPC bilayers obtained on GPUs versus experimental results (structural i.e., densities and electron density profiles, order parameters (S_{CD}) and electrostatic properties i.e., dipole potential and water dipole moments, as well as orientational properties of water) as the membranes significantly grow in size.

Specifically, the goals of this contribution are as follows. First, in Section II we discuss our implementation of PME algorithm for GPUs, noting the differences of the current implementation compared to previous approaches. In Section III we present accuracy and performance of simulations of three different lipid bilayer membranes (DMPC) with similar structures but different dimensions. In Section IV we discuss various properties of the model DMPC bilayer predicted from GPU-based molecular dynamics simulations using flexible non-polarizable force field for lipids and solvent. Section V concludes our contribution and discusses future prospects. As previous applications of GPU technology for molecular simulations have involved proteins in explicit and implicit solvent, *ab initio* quantum chemistry, and numerous algorithmic implementations for optimizing GPU calculations²⁵⁻³⁶, biological membranes and bilayers have not been represented in this set of applications. Thus, this study presents an initial foray into GPU-based simulations of membranes and integral membrane protein systems.

II. METHODS

We perform the membrane simulations on a single GPU using the GPU-based code FEN ZI (*yun dong de FEN ZI* in Mandarin or *moving MOLECULES* in English), an MD code enabling simulations at constant energy (NVE), constant temperature (NVT), and constant pressure and temperature (NPT). FEN ZI uses a modified version of the CHARMM force field⁴¹ in terms of force field functional forms and measurement units. The entire MD simulation (i.e., intra-molecular and long range potentials including PME) is performed on the GPU.

A. MD Models

The intra-molecular potential, which includes bonds, angles, including the Urey-Bradley summation, and dihedral, is as follows:

$$V_{intra} = \sum_{\text{bonds}} K_b(r - r_0)^2 + \sum_{\text{angles}} K_a(\theta - \theta_0)^2 + \sum_{\text{Urey-Bradley}} K_{UB}(S - S_0)^2 + \sum_{\text{dihedrals}} K_\phi(1 + \cos(n\phi - \delta)) + \sum_{\text{impropers}} K_\omega(\omega - \omega_0)^2 \quad (1)$$

where K_b , K_a , K_{UB} , K_ϕ , K_ω are the bond, angle, Urey-Bradley, dihedral and improper force constants, and r_0 , θ_0 , S_0 , δ , ω_0 are their corresponding equilibrium distances.

Non-bonded Lennard-Jones interactions are modeled using a standard 6-12 dispersion-repulsion potential:

$$V_{LJ} = \sum_{i,j}^{pairs} \left(4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \right) \quad (2)$$

where ϵ_{ij} and σ_{ij} are the potential well depth and the van der Waals radius, respectively, used in the Lennard-Jones potential.

Long range electrostatic interactions are computed using the Ewald summation method. In the Ewald summation method the electrostatic interactions are divided into the direct space energy (E_{dir}), the reciprocal space energy (E_{rec}), and the self energy (E_{corr}) contributions to the total energy, depending on the distance of the interaction³⁹. The three contributions are:

$$E_{dir} = \sum_{i=1}^{N-1} \sum_{j>i}^N \frac{q_i q_j \text{erfc}(\beta r_{ij})}{r_{ij}} \quad (3)$$

$$E_{rec} = \frac{1}{2\pi V} \sum_{\vec{m} \neq 0} \frac{\exp(-\pi^2 \vec{m}^2 / \beta^2)}{\vec{m}^2} S(\vec{m}) S(-\vec{m}) \quad (4)$$

$$E_{corr} = -\frac{1}{2} \sum_{(i,j) \in \text{Excl}} \frac{q_i q_j \text{erf}(\beta |r_i - r_j|)}{|r_i - r_j|} - \frac{\beta}{\sqrt{\pi}} \sum_{i=1}^N q_i^2 \quad (5)$$

where β is the Ewald parameter and $\vec{m} = (m_1, m_2, m_3)$ are reciprocal space lattice vectors, V is the volume of the unit cell in the reciprocal space, and $S(\vec{m})$ is the lattice structure factor given by:

$$S(\vec{m}) = \sum_j q_j \exp(i \vec{m} \cdot \vec{r}_j)$$

The solvent is treated explicitly. Water bond, angle, and charge parameters are transferred directly from the SPC/Fw model of Wu and co-workers⁴². DMPC is represented using the CHARMM c27r force field⁴³. We acknowledge that the current combination of solvent and biomolecule force fields have not been rigorously validated with decades of simulation studies; however, as the current implementation of FEN ZI lacks bond constraint algorithms, we opted to use a well-parameterized flexible water force field in conjunction with a well-parameterized lipid force field with the intention to assess the validity of the combination of force fields based on long time-scale molecular dynamics simulations enabled by GPU technology. The results presented in this paper indicate that this combination of fully flexible models is indeed adequate for modeling fully hydrated bilayer systems (as will be discussed in the Results section of this paper).

To perform constant temperature molecular dynamics simulations, FEN ZI maintains constant temperature using velocity reassignment. The velocity reassignment is performed every 20,000 time steps to set the system temperature. We acknowledge that the current method based on velocity reassignment does not reproduce the canonical ensemble. To address this issue, we are currently integrating the Nosé-Hoover thermostat⁴⁴ in the code. FEN ZI performs molecular dynamics simulations at constant pressure by using the Andersen barostat⁴⁵, wherein the Hamiltonian of the system includes terms arising due to fluctuations in the volume.

Last but not least, FEN ZI performs energy minimization of newly built and unminimized structures by using the hybrid combination of conjugated gradient and steepest descent methods. The conjugated gradient coefficient is calculated via the Fletcher-Reeves formula⁴⁶. Although efficient, the conjugate gradient method is unstable with respect to numerical perturbations and precision. Under finite precision assumptions, the conjugate vector may not preserve conjugation and could accumulate error over several steps of minimization⁴⁷. As a result the final structure is not fully minimized by the conjugated gradient method alone. The method of steepest descent, although slower, is applied to the partially minimized structure until the residual is close to minimal energy. The switch from

one method to the other is performed based on the total energy of the system which is monitored while progressively decreasing during the minimization.

B. GPU Code Implementation

We use CUDA for our code implementation. Bond-, angle-, and dihedral interactions are each handled by a unified kernel that evaluates the potentials by iterating through the list of all atoms bonded to or involved in an angle with an atom i , the kernel accumulates the bonded forces. Unlike non-bonded lists, the bond-, angle-, and dihedral lists never require updating, so they are constructed once on the CPU at the start of the simulation and then copied to the GPU. Individual threads on the GPU are assigned to individual atoms which iterate through the list of atoms they are bonded to. This approach is similar to other all-atom bonded force calculations included in our earlier implementation of this MD code⁴⁸ as well as in other MD codes that perform the bonded force calculations entirely on GPUs, e.g., AMBER⁴⁹, HOOMD-blue⁵⁰, OpenMM⁵¹, and ACEMD⁵².

Non-bonded interactions (i.e., Lennard-Jones and direct space electrostatic terms) are handled by two kernels, one for building the nonbond neighbor list and one for computing the interactions. FEN ZI uses the cell-based approach for building the nonbond neighbor list. The entire domain of atoms is decomposed into regular cubic cells of edge length equal to r_{cut} where r_{cut} is the cut-off radius used in the energy calculation. The neighbor list is constructed using the Verlet list approach⁵³, in which, for each atom, a pairlist is constructed containing all atoms within a cut-off radius r_{list} where $r_{list} > r_{cut}$. The cut-off radius r_{list} is used to construct and update the pairlist; the region between the two cut-off radius r_{list} and r_{cut} serves as a buffer. The nonbonded Verlet list and its corresponding update criteria are used by other MD codes such as the CHARMM code on CPU⁴¹ to efficiently calculate short range nonbond interactions. On the other hands, other GPU codes such as ACEMD⁵² use cell-based computation of nonbond forces wherein the domain is divided into smaller cells

of size, $\frac{1}{2}r_{cut}$. The cells that are in the immediate neighborhood and those further away are handled by different kernels and the code implementation does not seem follow a Verlet list approach for nonbond computations. In our implementation, while building the neighbor list for each atom following the Verlet approach, only the atoms within the current cell and 26 neighboring cells in 3-dimensional space need to be searched for atoms within the cutoff distance. The cells themselves are updated efficiently by carefully keeping track of movement of atoms within the cells and employing atomic intrinsics where possible. Each thread iterates through the list of neighbors for a single atom i and accumulates the interactions between i and all its neighbor list entries. The texture cache, which speeds up reading from global memory locations that are not contiguous, is used for reading the coordinates of the neighbor atoms. Switched force forms are used for the Lennard-Jones potential to ensure that both energies and forces go smoothly to zero at the cutoff r_{cut} .

The Smooth Particle Mesh Ewald (PME) component of the Ewald summation method (E_{rec}) requires us to deal with location of charges. The entire procedure involves: (1) spreading the charges within a neighboring volume of $4 \times 4 \times 4$ cells for each charge in order to obtain a 3-dimensional charge matrix, (2) computing the inverse 3D FFT of the matrix, (3) multiplying the charge matrix by pre-computed structure constants, (4) computing the forward FFT of the product, and (5) summing the entries of the matrix to obtain energy and forces³⁹. The charge spreading is performed in one kernel and the list updating is done in a second kernel. The FFT computations, i.e., inverse and forward FFTs, are performed using the CUDA FFT library. The multiplication of the charge matrix by pre-computed structure constants and the summation of the entries of the matrix to obtain energy and forces are performed by two separate kernels.

Of these five steps, the charge spreading is one of the most compute intensive. In Ref.⁵², where another example of MD code with PME for GPUs is presented, this step is implemented by a three-step process which involves placing the charges on lattice points with utmost one charge per point (with extra charges considered separately), accumulating the impact of charges for each points, and finally accumulating the effects of the extra charges. Our implementation differs from the implementation in Ref.⁵² since we accumulate impact of charges based on a local list of charges for each lattice point. Our charge spreading on GPU is performed by maintaining a local neighbor list for each lattice point, followed by an efficient update of list caused by the movement of particles. In general, the charge contained in each atom is spread within a neighborhood of $4 \times 4 \times 4$ lattice points around the particle (Fig. 1). The spreading of charges is done with the help of the cardinal B-spline according to the summation as follows:

$$Q(k_x, k_y, k_z) = \sum_{i=1}^N \sum_{n_1, n_2, n_3} q_i M_4(u_{xi} - k_x - n_1 K_x) \times M_4(u_{yi} - k_y - n_2 K_y) \times M_4(u_{zi} - k_z - n_3 K_z) \quad (6)$$

When a charge moves due to dynamics of the system, as shown in Figure 2 where a charge moves from Point 2 to Point 2', we can identify three regions of the lattice: (1) a region from which the charge is removed from the neighbor lists (light gray region in Figure 2), (2) a region whose neighbor lists gain the charge (dark grey region in Figure 2), and (3) a region whose neighbor lists are not impacted (white region in Figure 2). The solid arrow indicates the charge movement; dash arrows indicate mapping between lattice points. With reference to the mapping, the lattice points gaining the charge impact are initially not aware of the charge movement and thus need to search through the global list of charges for the information when an update takes place. This search is time demanding. At the same time, we observe that the movement of a charge produces an equal number of lists gaining the charge and lists losing the charge. The latter lists are associated to lattice points that are aware of the destination points in a one-to-one mapping schema showed in Figure 2. We can benefit from this one-to-one mapping to reduce the time for the update of the lists by having the threads of the points losing the charge updating the lists of the points gaining the charge. This will prevent the lattice points gaining the charge impact to explore the whole global list of charges.

In case multiple charges from different lattice points move into the neighborhood of the same lattice point, then the neighbor list of that point is updated by multiple threads. In Figure 3 the local neighbor list of a point is updated by two different threads due to the movement of two different charges (Charges 2 and 1). Since this update requires the addition of two charges by different threads to the same neighbor list, this is performed with the help of thread safe integer atomic intrinsics. The outline of the code listing for the CUDA kernel to update the lattice neighbors is shown in Listing 1. To minimize the number of floating-point calculations, we pre-compute values of B-Spline function, i.e., the structure constants, and store them in the GPU constant memory.

III. ASSESSING ACCURACY AND PERFORMANCE

Lipid bilayer membranes are studied in this paper. At the same time they serve as benchmarks to assess accuracy and performance for our GPU codes. We use CHARMM⁴¹ as our reference for assessing both FEN ZI accuracy and performance. When addressing accuracy, the question we want to answer is whether the overall simulation behavior matches the results of the same simulation when using CHARMM. When addressing

performance, the question we want to answer is how much simulated time (e.g., in nanoseconds or microseconds) can be performed in one day and whether simulations scale in both time and space scales.

A. Hydrated DMPC Bilayers

We consider three different lipid bilayer membranes (DMPC) with similar structures but different dimensions (Figure 4), each one four times larger than the previous:

- Small (DMPC 1×1): $46.8\text{\AA} \times 46.8\text{\AA} \times 76.0\text{\AA}$, 17,004 atoms (14,096 bonds, 19,108 angles, and 22,536 dihedrals), 2,836 explicit water molecules, 72 DMPC molecules
- Medium (DMPC 2×2): $93.6\text{\AA} \times 93.6\text{\AA} \times 76.0\text{\AA}$, 68,484 atoms (56,696 bonds, 76,588 angles, and 90,144 dihedrals), 11,500 explicit water molecules, 288 DMPC molecules
- Large (DMPC 4×4): $187.2\text{\AA} \times 187.2\text{\AA} \times 76.0\text{\AA}$, 273,936 atoms (226,784 bonds, 306,352 angles, 360,576 dihedrals), 46,000 explicit water molecules, 1152 DMPC molecules

The systems are simulated with a constant surface area per lipid of $60.8\text{\AA}^2$. This is within the uncertainty of the experimentally measured value, $60.6 \pm 0.5\text{\AA}^2$ ⁵³. Molecular dynamics simulations of the three membranes are performed with velocity reassignment every 20000 steps to maintain system temperature of 298K. Although slightly lower than that used in previous molecular dynamics studies of similar systems, this temperature is above the gel-liquid transition temperature; as evidenced by the results presented later, the combination of force fields at this temperature is able to recapitulate most of the structural properties of DMPC bilayers observed previously by theory, simulation, and experiment. We comment that the actual phase transition temperature of the combination of the c27r lipid force field and SPC/Fw water model remains outstanding, though again, our results indicate that with respect to structural properties, the force field models represent DMPC bilayers above the transition. A velocity Verlet integrator with 1 femtosecond (fs) timestep (to achieve numerical stability with flexible bonds) propagates the system.

B. Accuracy

Clearly, even the fastest MD code has little value if the results it produces are not accurate. According to Allen and Tildesley⁵⁴, in an accurate MD simulation a plot of the energy fluctuations versus timestep size should follow an approximately logarithmic trend. To ensure that our simulation results are accurate, we measured and analyzed the energy fluctuations as a function of the timestep size for two different computation precisions (i.e., single and double precision) and two sets of cutoff values. In Figure 5, the total energy fluctuation is plotted for five different timestep sizes (i.e., 0.125 fs, 0.25 fs, 0.5 fs, 1 fs and 2 fs). All the tests for the considered timesteps, were performed on a well equilibrated initial structure of the smaller membrane (DMPC 1×1) at 298K. The standard deviation of the energy was computed over ten consecutive 2 ps intervals. The means of the standard deviations for the ten different intervals are plotted for each timestep size. In Figure 5 we observe that, for large timestep sizes (larger than 0.5 fs), the fluctuations in total energy of MD simulations are proportional to the timestep size independently from the type of precision and cutoff values. This conclusion supports the accuracy of our simulations performed using 1 fs timestep size, single precision, and with cutoff values as follows: cuton=7 Å, cutoff=8 Å and cutnb=9.5 Å. On the other hand, we observe a different behavior for step size less than 0.5 fs. This is consistent with results previously presented and discussed in Ref.⁴⁸. Results for the same calculations on the same system using a double precision CHARMM executable on a single core are also shown for timesteps of 0.25, 0.5, 1.0, and 2.0 fs in Figure 5. For a timestep of 2.0 fs, we observe that double precision

CHARMM exhibits a larger fluctuation in total energy compared to FEN ZI; this is consistent with the fact that the test is performed using flexible water model. The quadratic dependence of energy fluctuations on timestep thus deteriorates at this higher time step, indicative of the onset of instability in the numerical integrator. At the lower timesteps, the double precision CHARMM executable follows the expected quadratic dependence of energy fluctuations on timestep. At a 1.0 fs timestep, both GPU and CPU exhibit a similar energy fluctuation; this is the timestep used in the current study and thus we believe that this demonstrates that the use of single precision in the present study is commensurate to using double precision CPU hardware. At timesteps smaller than 1.0 fs, we observe that the use of single precision on the GPU does lead to deterioration of integrator stability, as expected.

To ensure that our simulation results are accurate, we also ran simulations with the smaller membrane system (DMPC 1×1) in equilibrium on CPU using the double-precision CHARMM code and on GPU using our single precision code. The simulations are performed in constant temperature. As pointed out by Friedrichs and co-workers²⁵, energy drift can be observed in long constant energy simulations and the drift can be easily overcome by maintaining constant temperature (e.g., using velocity reassignment which effectively changes the energy state of the system at specified intervals, thereby eliminating the drift for production simulations).

Figure 6 plots the temperature (a function of kinetic energy), bond energies (i.e., bond-, angle, Urey-Bradley-, improper-, and dihedral energies), and non-bond energies (i.e., VDW, electrostatic, and PME) over a 3 ns simulation time for the small membrane. The several quantities (i.e., temperature and energies) fluctuate, as expected, around the same average value for both the double precision CPU simulation and single precision GPU simulation. We observed that FEN ZI initial energy (step 0) matches the CHARMM initial energy. During the initial phase of the simulation, before the system reaches the equilibrium, we also observed some energy drifting due to the different thermostats used by the two codes (i.e., CHARMM uses Langevin thermostat while FEN ZI uses velocity reassignment). Eventually the two simulations converge, indicating that FEN ZI does in fact produce results consistent with CHARMM and the use of double precision is not needed on GPUs for this type of simulations.

C. Performance

We compare the CHARMM performance versus FEN ZI performance with the same input files for the small DMPC 1×1 - i.e., coordinates, topology, and input parameters. For both CHARMM and FEN ZI, we ran 10,000 steps of a NVE MD DMPC 1×1 simulation (constant energy) using a r_{cutoff} of 8 Å with a buffer cutoff r_{list} for the list updates of 9.5 Å. Each timestep is 1 fs long. CHARMM simulations were run on 1, 2, 4, and 8 cores of an Intel Xeon X5355 with speed 2.6 GHz and 8 GB of memory. CHARMM uses MPI for distributing the membrane domain across the cores and was optimized for the specific nodes. We did not scale CHARMM beyond 8 cores of a single node because the cluster used was not provided with InfiniBand and thus the performance was very poor across nodes. For our GPU simulations, we used both a GTX 480 GPU (Fermi) with 480 cores and 1.5 GB memory and a C2050 GPU (Fermi) with 448 cores and 3 GB memory. The performance testing and scalability analysis presented in this paper were performed on the current generation of NVIDIA Fermi GPUs. The suite of optimizations implemented in FEN ZI (i.e., optimal use of 64KB of L2 cache, GPU thread block assignments and register usage) are applicable to the latest generation of NVIDIA GPUs. The code can still run on older generations by turning off some of the optimizations.

Figure 7 shows the number of nanoseconds per day for the double-precision parallel CHARMM code optimized for 1, 2, 4, and 8 CPU cores versus the number of nanoseconds

per day of FEN ZI on one single GPU. The performance values presented are the average values computed over three repeated simulations. The standard deviation is not reported because it is close to zero. The figure shows how the speedup of one single-precision GPU is up to 20.5 ns/day on a GTX 480, i.e., more than 10X faster than a 8-core, double-precision node. Note that all 8 cores have to be used to run CHARMM for its best performance of 1.9 ns/day. We acknowledge that the CHARMM simulation was performed using double precision arithmetic, thus introducing some performance advantage for the single precision GPU. We stress that the test here is to compare standard hardware platforms commonly used for molecular dynamics simulations with the current GPU technology.

The scalability of FEN ZI was measured by running 10,000 steps of a NVE MD simulations for the three membranes, i.e., DMPC 1×1, DMPC 2×2, and DMPC 4×4, using FEN ZI on GPU. Note that we did not measure the scalability of CHARMM for this comparison. Overall FEN ZI enables multi-nanosecond MD simulations per day with all three membranes. More in particular, for the smaller membrane (DMPC 1×1), our MD code running on GPU achieves simulation rates of up to 20.5 million MD steps with 4.19 ms per step (20.5 nanoseconds per day with a MD step size of 1 fs) on a GTX 480 and up to 15.5 million MD steps with 5.55 ms per step (15.5 nanoseconds per day with a MD step size of 1 fs) on a C2050. For the medium membrane (DMPC 2×2), our MD code running on GPUs achieves simulation rates of up to 6.3 million MD steps per day with 13.7 ms per step (6.3 nanoseconds per day with a MD step size of 1 fs) with a GTX 480 and up to 4.65 million MD steps with 18.56 ms per step (4.65 nanoseconds per day with a MD step size of 1 fs) on a C2050. Last but not least, for a large membrane (DMPC 4×4) we reach up to 1.5 million MD steps with 57.36 ms per step (1.5 nanoseconds per day with a MD step size of 1 fs) on a GTX480 and up to 1.1 million MD steps with 77.12 ms per step (1.1 nanoseconds per day with a MD step size of 1 fs) on a C2050. The overall scalability results are summarized in Figure 8.

Overall, FEN ZI allows us to simulate larger membranes, larger than simple regions, over a longer simulated time interval in a significantly shorter turnaround time.

IV. RESULTS AND DISCUSSION

In this section we consider various structural and electrostatic properties for each of the DMPC 1×1, 2×2, and 4×4 systems described in Section III A; again, the nomenclature describes the system size in terms of the smallest building block consisting of a solvated bilayer patch of two leaflets, each consisting of 36 DMPC lipid molecules. Total simulation times of 100 ns, 100 ns, and 64 ns are achieved for the DMPC 1×1, 2×2, and 4×4 systems, respectively. For the purposes of the analysis in this section, we average properties over all but the first 2 ns, which is taken as equilibration.

Most properties in this section are determined as a function of the z -position, where z is the direction normal to the water-lipid interface. Furthermore, the symmetry of these systems allows us to effectively double the amount of sampling across the water-lipid interface by considering the average properties of the two interfaces as a function of the absolute distance in the z -direction from a reference value. That is, z is recast as $z = |z' - z_{\text{ref}}|$ where z' is the original position of the particle in the system in simulation coordinates and z_{ref} represents the z -component of center of mass for the bilayer. In all cases we consider the x and y directions to be isotropic; that is, any variability in the lateral properties of the lipid bilayer are averaged out.

A. Atomic and Electron Densities

Standard structural properties of lipid bilayers are the atomic and electron densities. The number densities for select head group atoms (nitrogen, phosphorus, oxygen atoms singly and doubly bonded to phosphorus, and carbonyl oxygen atoms) are shown in Figure 9 for the small (DMPC 1×1), medium (DMPC 2×2), and large (DMPC 4×4) systems. For every DMPC molecule, the number of each type of oxygen shown is twice that of each nitrogen and phosphorus, resulting in oxygen peaks with twice the area of those for nitrogen and phosphorus. For reference, the tail of the number density for water is also shown. In all systems, the water density drops sharply, and approaches zero near the position of the carboxylic oxygens of the ester group. The width of the headgroup atom profiles suggests the relative variation in the z -position of these headgroup atoms. We observe similar peak widths for the small and large system profiles, while the medium system profiles are less broad. The peak positions are consistent with those from previous simulation studies of polarizable and non-polarizable systems^{43,55}.

While these number density profiles cannot be directly related to an experimental measurements, they reflect features present in the electron density profiles shown in Figure 10. An experimental electron density profile was previously generated in Ref.⁵⁴ by fitting Gaussian functions to match experimental form factors. The experimentally-derived electron density profiles are shown along with the simulation profiles computed from each system. All simulation profiles agree reasonably well with the experimental profile. The agreement of the profiles computed in this study with experiment is comparable to that found in other studies⁵⁵.

Finally, we note that the simulation profile is slightly reduced relative to the experimental profile in the bulk water region. This is expected since the water density calculated in this study is slightly reduced relative to experiment.

B. Orientation of Water

The orientation of water is calculated from the angle formed between the permanent dipole vector of water (the HOH angle bisector) and the z -axis. In keeping with previous studies⁵⁵ we consider the average cosine of this angle as a function of the z -distance. From this definition, orientation of water can be interpreted as follows. Values of $\langle \cos \theta \rangle = 0$ suggest that water orientation is isotropic; this condition also holds if water is oriented such that the permanent dipole of water is perfectly parallel with the water-lipid interface. The condition in which $\langle \cos \theta \rangle = \pm 1$ corresponds to perfect alignment of the permanent dipole along either the positive or negative z axis. While such perfect alignment of all water within a given z -position is unlikely, fractional values of this quantity can suggest whether there is a net orientation of water molecules and give a general sense of the direction of this net orientation. The water orientation profiles are shown in Figure 11. In the bulk water region ($z > 25$), $\langle \cos \theta \rangle = 0$ suggesting water orientation is isotropic. Approaching the headgroups, however, water demonstrates a preferential orientation such that the hydrogen atoms are pointed away from bulk water; that is, the permanent dipole of water is preferably oriented along the negative- z axis. Moving further into the lipid bilayer ($z < 15\text{\AA}$), water re-orientes such that it becomes preferentially aligned with the positive z -axis. Although there is a stronger net orientation in this region than at the headgroups, the number of water molecules penetrating the lipid bilayers is limited; this limited sampling is particularly evident at depths $z < 7\text{\AA}$ where the orientation varies significantly. These observations are qualitatively similar to those in Ref.⁵⁵.

C. DMPC Orientation and Dynamics

Orientation of the lipid headgroups can be assessed through the angle made between the phosphorus to nitrogen vector ($\vec{PN}$) and the normal to the bilayer-water interface ($\vec{z}$). We consider the distribution of this angle in Figure 12. It is clear from these profiles that $\vec{PN}$ samples a broad range of angles with the bilayer normal with the most probable range being $\sim 70-90^\circ$ (with a peak height occurring at about 81). This result is slightly reduced relative to previous studies of DMPC using the CHEQ and C27r force fields⁵⁵ well as the GROMOS force field⁵⁶ which predict a most probable value of ~ 90 . We note that previous studies of DPPC demonstrate favorable orientation of this vector at angles of $\sim 80-90^\circ$.^{57,58}

The order of lipid tailgroups is investigated through the deuterium order parameters, which are calculated as a function of the position along the alkyl chain. The deuterium order parameter, S_{CD} is calculated as

$$S_{CD} = \langle P_2(\cos\theta) \rangle \quad (7)$$

where $P_2(\cos\theta) = (1/2)(3(\cos\theta)^2 - 1)$ and describes the angle between a given CH vector and the interface normal. This quantity can be related to the S_{zz} component of the NMR quadrupolar splitting tensor. We show the $sn-1$ and $sn-2$ order parameters calculated for each system, as well as the experimental values in Figure 13. Experimental values for the 2R and 2S hydrogen atoms are available and distinguished in this figure. In all systems, the order profiles show qualitative agreement with the experimental profile (although the $sn-2$ | S_{CD} are generally higher than experiment); this is consistent with previous studies of DMPC bilayers using the C27r force field.⁵⁵

D. Water Dipole Moment Profile

One noteworthy property of water is variation of dipole moment across the water-lipid interface. As has been demonstrated by models in which polarizability is explicitly treated,⁵⁵ the average dipole moment of water decreases upon transitioning from bulk water ($e \approx 80$) to the lipidic interior of DMPC ($e \approx 2$). Such variation of molecular dipole moment cannot be captured by *rigid*, fixed charge models. However, a flexible, fixed-charge model (such as SPC/Fw as used in the current study) can model variation of dipole moment across such interfaces through the bending of water's internal angle. The profiles for the average molecular dipole moment of water are shown in Figure 14. In the bulk water region ($z = 25-35$ Å), the average dipole moment of water is 2.39 and shows little deviation. Approaching and entering the DMPC region, the dipole moment of water is monotonically reduced by approximately 0.2 D. This reduction is attributed to a relaxation of the HOH bond angle (the bond angle increases), as was previously noted in the context of simulations of SPC/Fw at the water-air interface⁴⁸.

E. Electrostatic Potential

The surface potential can be calculated by twice integrating the z -dependent charge density as

$$\Phi(z) = -\frac{1}{\epsilon_0} \int_0^z \int_0^{z'} \rho(z'') dz'' dz' \quad (8)$$

Here, $z = 0$ denotes the position of the center of the bilayer, ϵ_0 is the permittivity of a vacuum, and $\rho(z)$ is the charge density taken over all atoms within a slab at a distance z from the bilayer center. This potential can be further decomposed into contributions arising from the water

$$\Phi_{\text{H}_2\text{O}}(z) = -\frac{1}{\epsilon_0} \int_0^z \int_0^{z'} \rho_{\text{H}_2\text{O}}(z'') dz'' dz' \quad (9)$$

and the lipid

$$\Phi_{\text{DMPC}}(z) = -\frac{1}{\epsilon_0} \int_0^z \int_0^{z'} \rho_{\text{DMPC}}(z'') dz'' dz' \quad (10)$$

where the charge densities due to water ($\rho_{\text{H}_2\text{O}}(z)$) and DMPC ($\rho_{\text{DMPC}}(z)$) are distinguished. Surface potential profiles (total and contributions from water and DMPC) are shown in Figure 15. We summarize the changes in total potential and contributions in Table I for each system studied. The total potential going from the bilayer to water is about 1 V and results are consistent among the different system sizes (spanning a range of only 40 mV). The contribution from the DMPC is on the order of 2.3 to 2.5 V, while the contribution from water is on the order of -3.5 to -3.4 V. Although the total contribution is on the order of previous predictions from nonpolarizable ($\sim 0.8\text{V}$) and polarizable ($\sim 0.9\text{V}$), contributions from water and lipid are larger in magnitude in this study than in the study by Davis *et al.*⁵⁵. We note that the simulations in the current study have been performed at constant volume and temperature, as opposed to constant surface area and pressure (*NPAT*) ensemble in previous work⁵⁵. Furthermore, the water model used in previous studies of nonpolarizable systems (TIP3P^{54,59}, SPC⁵⁹) has been replaced by (SPC/Fw⁴²) in the current study. These factors lead to nontrivial differences in some calculated properties. Most relevant to surface potential, differences in water orientation and headgroup orientation ($\langle \vec{p}_N \rangle$) may contribute to the enhanced magnitude in water and lipid contributions.

It is interesting to note that the water contribution to the surface potential is relatively constant within the bilayer interior, despite the presence of an orientational preference of the water dipoles (Figure 11). Although the slightly reduced molecular dipole moment of water within the lipid (Figure 14) may influence the constant potential in this region to some extent, this observation is more likely an effect of the reduced water density. That is, the dipole density becomes negligible within the bilayer interior. We explicitly demonstrate the effect of low density on the oriented water molecules within the lipidic region in Figures 11b,d,f for the 1×1 , 2×2 , and 4×4 systems, respectively. Here, the average water orientation is scaled by the water density at the given position. For the purposes of this analysis, the calculated water density profiles ($\rho(z)$) were fitted to an error function of the form:

$$\rho_{\text{fit}}(z) = \frac{\rho_{\text{bulk}}}{2} + \frac{\rho_{\text{bulk}}}{2} \text{erf}\left(\frac{z - z_0}{w}\right) \quad (11)$$

where ρ_{bulk} is the average bulk number density of water, z_0 is the inflection point of this

profile that corresponds to the point in which $\rho(z) = \frac{1}{2}\rho_{\text{bulk}}$, and w is the intrinsic width of this profile. In order to consider a scaling from 0 to 1 (representing the relevant extremes of no water presence to bulk water density), we normalize the fitted profile by the density of bulk water:

$$\xi(z) = \frac{\rho_{\text{fit}}(z)}{\rho_{\text{bulk}}} = \frac{1}{2} + \frac{1}{2} \text{erf}\left(\frac{z - z_0}{w}\right). \quad (12)$$

The product $\xi(z) \langle \cos \theta(z) \rangle$ shown in the right panels of Figure 11 suggests the influence of the orientation of water dipoles on the surface potential at a given z -position when the relative abundance of water is considered. Indeed, we conclude that the nontrivial alignment

of water dipoles in the range $z = 10 - 15 \text{ \AA}$ will not significantly influence the surface potential as a result of low water density in this region.

The significant drop in the water contribution to surface potential occurs at distances corresponding to the onset of negative values for $\langle \cos \theta \rangle$ in the water orientation (or scaled orientation) profiles. This is not entirely surprising since the scaled density profiles are essentially scalar multiples of the water dipole densities, where the scaled orientation profiles would be multiplied by the ρ_{bulk} and $\langle \mu(z) \rangle$. Although we neglect the variation of $\langle \mu(z) \rangle$ (previously demonstrated in Figure 14) by stating the water dipole density is a scalar multiple of the scaled orientation, this assumption is appropriate over the range in which ρ_{fit} takes on non-zero values. Relative to the Davis and co-workers⁵⁵ study, the minimum in the orientation profile is slightly more negative, indicating a stronger alignment of dipoles along the bilayer normal. This, coupled with an increased magnitude of dipole moment (2.39 D for SPC/Fw compared to 2.35 for TIP3P), would give rise to the observed increase in magnitude of the water contribution to the surface potential.

V. CONCLUSIONS

GPU simulations enable studying membrane-bound protein receptors beyond the current state-of-the-art simulations that only consider small membrane patches. Our code FEN ZI includes advanced GPU algorithms for Molecular Dynamics (MD) simulations of large membrane regions using explicit solvent and Particle Mesh Ewald (PME) method for treating the conditionally convergent electrostatic component of the classical force field. Because of our general design and implementation approach, FEN ZI is full-fledged general purpose MD package based on canonical force fields and PME for GPU enabled simulation of a broad class of molecular systems. FEN ZI achieves simulation rates of up to 20.5 nanoseconds per day with a MD step size of 1 fs for a small DMPC membrane of 17,004 atoms (versus 1.9 nanoseconds per day of CHARMM on an 8-core system), up to 6.3 nanoseconds per day with a MD step size of 1 fs for a medium DMPC membrane of 68,484 atoms, and up to 1.5 nanoseconds per day with a MD step size of 1 fs for a large DMPC membrane of 273,936 atoms; all membranes were in explicit solvent.

In this paper, we present a comparison of structural and electrostatic properties computed using FEN ZI with results from experiment and state-of-the-art all-atom molecular dynamics simulations. We see excellent agreement between computed S_{CD} and electron density profiles with experiment. Surface potential across the lipid-water interface is calculated to be -1.0 V , which agrees reasonably well with predictions from non-polarizable and polarizable force fields (approximately -0.8 V and -0.9 V respectively)⁵⁵. Individual contributions to this potential from water and lipid are enhanced relative to previous studies, which is likely a result of using a different water model as well as changes in the orientation of water and headgroups at the interface. Of significant note is the fact that simulation results for structural and electrostatic properties were not significantly influenced by the size of the bilayer studied (72 vs. 288 vs. 1152 DMPC molecules). Though certain properties related to the fluctuation dynamics of the bilayer will be different depending on the system size, we reserve analysis of such behavior for future studies. In conclusion, we emphasize that our contribution demonstrates the utility of GPU technology in computational membrane biophysics and physiology. Continuing work in our laboratory focuses on exploring several fundamental aspects of membrane biophysics. These include the nature of peptide insertion (with renewed interest in the mechanisms of peptide insertion coming to light as in Reference⁶⁰), the nature of oxidative lipid damage and the mechanisms of aggregation and dynamics of damaged lipids in such systems⁶¹, the role of meso-scale domains in membrane structure and physiology, as well as studies of the interactions of multiple peptides in membrane bilayers at fully atomistic detail. We believe that routine GPU-enabled

simulations on the meso-scale will provide unprecedented insights into the biophysics of this important class of biomacromolecules.

Acknowledgments

The authors gratefully acknowledge support from the National Institute of Health sponsored COBRE (Center of Biomedical Research) Grant #P20-RR017716 at the University of Delaware (Department of Chemistry and Biochemistry) and P20-RR015588 (Department of Chemical Engineering); the National Science Foundation, with Grant #0941318 and Grant #0922657, the U.S. Army, with Grant ARO #54723-CS, and the NVIDIA University Professor Partnership Program. B.A.B. acknowledges additional support from a Graduate Fellows Award at the University of Delaware.

Further information and support for FEN ZI is available from <http://gcl.cis.udel.edu/projects/fenzi/index.php>

APPENDIX A: PERFORMANCE OF DHFR BENCHMARK

DHFR systems rather than DMPC systems are usually used as benchmarks for comparing performance on GPUs. DHFRs are not the scope of this paper, nevertheless we report performance for DHFR for the sake of comparison with other GPU codes. A detailed analysis of the performance for DMPC and DHFR systems from the computer science point of view are reported in Ref.⁶².

To compare FEN ZI performance with other codes performing PME entirely on GPU (i.e., ACEMD and AMBER), we ran the MD simulation of a well-known benchmark such as the DHFR system (23K atoms with 16,569 bonds, 11,584 angles, and 6,701 dihedrals) using initial parameters that are similar, when possible, to the parameters used by the other codes in their webpages, i.e., 8 Å nonbond cutoff, a buffer cutoff of 9.5 Å, 10,000 MD steps in an NVE ensemble, and with both 1 fs and 2 fs time step. Figure 16 shows the performance of DHFR MD simulations with FEN ZI on GTX 480 and C2050 (Fermi chip).

Performance results on ACEMD and AMBER are presented in Ref.⁶³ and in Ref.⁴⁹ respectively. While comparing these results with our results, we observe that FEN ZI gives similar performance as AMBER (i.e., according to ⁴⁹: AMBER reaches 21.11 ns/day on a C2050 with a MD step of 2 fs). ACEMD is faster than FEN ZI (i.e., from our testing of ACEMD on GTX 480, the code reaches 37.24 ns/day with a MD step of 2 fs). Note that our MD algorithm is different than the ACEMD algorithm: as outlined in Ref.⁶², we deal with the PME charge spreading in a significantly different way than ACEMD. Moreover, the nonbond interactions in FEN ZI still use exclusion and non-bond lists as in codes such as CHARMM, while ACEMD does no longer refer to these data structures in Ref.⁵². GROMACS performance in Ref.⁶⁴ indicates values below 20 ns/day for the DHFR system on a C2050 GPU, however the webpage does not include important information such as cutoff values and other parameter settings. Thus we cannot fairly compare these values with FEN ZI performance.

Last but not least, in this paper we do not compare performance of FEN ZI with the GPU-based NAMD⁶⁵ since the current version of NAMD for GPUs does not perform PME on GPUs (thus using different algorithms for the electrostatic interactions) and computes the bonded force interactions on CPU. The performance details for NAMD for the relevant DHFR benchmark under a variety of configurations is available from the code webpage⁶⁶.

Listing 1: LatticeUpdate CUDA kernel

```
__global__ void Lattice_Update(){//lattice update kernel

2 float x, y, z;
```



```

lattice_coordinates(x, y, z, lattice); //copy coordinates of current lattice into x, y, z

4 for (i=0; i<num_charges[lattice]; i++){

chargeid = get_lattice_neighbor(lattice, i); //get the i-th neighbor of lattice

6 int3 disp = displacement[chargeid]; //displacement of 'chargeid' integral number of
lattices.

boolean flag = 0;

8 float3 dr = compute_nearest_image(r, coordinates[chargeid]);

//each thread updates one lattice.

10 if (((dr.x>=8)|| (dr.y>=8)|| (dr.z>=8)|| (dr.x<=-4)|| (dr.y<=-4)|| (dr.z<=-4))&&(!flag)){

//if charge displacement is greater than 8 units or less than -4

12 x1 = x + disp.x; y1 = y + disp.y; z1 = z + disp.z; flag = 1;}

//otherwise if any of the following is true, update only those affected.

14 if ((dr.x>4)&&(!flag)){

x1 = r.x + 4; y1 = y + disp.y; z1 = z + disp.z; flag = 1;}

16 if ((dr.x<0)&&(!flag)){

x1 = x - 4; y1 = y + disp.y; z1 = z + disp.z; flag = 1;}

18 if ((dr.y>4)&&(!flag)){

x1 = x; y1 = y + 4; z1 = z + disp.z; flag = 1;}

20 if ((dr.y<0)&&(!flag)){

x1 = x; y1 = y - 4; z1 = z + disp.z; flag = 1;}

22 if ((dr.z>4)&&(!flag)){

x1 = x; y1 = y; z1 = z + 4; flag = 1;}

24 if ((dr.z<0)&&(!flag)){

x1 = x; y1 = y; z1 = z - 4; flag = 1;}

26 if (flag){

lattice_to_update = compute_lattice_id(x1, y1, z1);

28 number_charges = atomicAdd(&num_charges[lattice_to_update], 1);

//add 'chargeid' to the lattice neighborlist of 'lattice_to_update'

30 add_lattice_neighbor(lattice_to_update, number_charges, chargeid); }

```

```

} //end for (i=0; i[<]num_charges[lattice]; i++)

32 global_synchronize_threads(); //synchronize all threads to avoid racing conditions.

Lattice_Clean(); //remove obsolete charges from the neighbor.

```

References

- [1]. Roux B, Allen T, Berneche S, Im W. Theoretical and Computational Models of Ion Channels. Quarterly Reviews of Biophysics. 2004; 37:15–103. [PubMed: 17390604]
- [2]. Wang L, Bose PS, Sigworth FJ. Using cryo-EM to measure the dipole potential of a lipid membrane. PNAS. 2006; 103:18528. [PubMed: 17116859]
- [3]. Stern H, Feller SE. Calculation of the dielectric permittivity profile for a nonuniform system: application to a lipid bilayer simulation. J. Chem. Phys. 2003; 118:3401.
- [4]. Allen TW, Bastug T, Kuyucak S, Chung S-H. Gramicidin A Channel as a Test Ground for Molecular Dynamics Force Fields. Biophysical Journal. 2003; 84:2159–2168. [PubMed: 12668425]
- [5]. Berneche S, Roux B. The Ionization State and the Conformation of Glu-71 in the KcsA K⁺ Channel. Biophysical Journal. 2002; 82:772–780. [PubMed: 11806919]
- [6]. Allen TW, Andersen OS, Roux B. Energetics of Ion conduction through the gramicidin channel. PNAS. 2004; 101(1):117–122. [PubMed: 14691245]
- [7]. MacCallum JL, Bennett WFD, Tieleman DP. Distribution of Amino Acids in a Lipid Bilayer from Computer Simulations. Biophysical Journal. 2008; 94:3393–3404. [PubMed: 18212019]
- [8]. Dorairaj S, Allen TW. On the thermodynamic stability of a charged arginine sidechain in a transmembrane helix. PNAS. 2007; 104:4943–4948. [PubMed: 17360368]
- [9]. Aliste MP, Tieleman DP. Computer Simulation of partitioning of ten pentapeptides Ace-WLXLL at the cyclohexane/water and phospholipid/water interfaces. BMC Biochem. 2005; 6:30. [PubMed: 16368010]
- [10]. Tieleman DP, MacCallum JL, Ash WL, Kandt C, Xu Z, Monticelli LM. Membrane Protein Simulations with a united atom lipid and all atom protein model: side chain transfer free energies and model proteins. J. Phys. Condens. Matter. 2006; 18:S1221. [PubMed: 21690838]
- [11]. Xu Z, Luo HH, Tieleman DP. Modifying the OPLS-AA force field to improve hydration free energies for several amino acid side chains using new atomic charge and an off-plane charge model for aromatic residues. J. Comput. Chem. 2007; 28:689. [PubMed: 17195160]
- [12]. Feller SE. Molecular dynamics simulations of lipid bilayers. Current Opinion in Colloid & Interface Science. 2000; 5:217.
- [13]. MacCallum JL, Tieleman DP. Computer Simulation of the distribution of hexane in a lipid bilayer: spatially resolved free energy, entropy, and enthalpy profiles. J. Am. Chem. Soc. 2006; 128:125. [PubMed: 16390139]
- [14]. Pandit SA, Bostick D, Berkowitz ML. Complexation of Phosphatidylcholine Lipids with Cholesterol. Biophys. J. 2004; 86(3):1345. [PubMed: 14990465]
- [15]. Berkowitz ML, Bostick DL, Pandit SA. Aqueous Solutions next to Phospholipid Membrane Surfaces: Insights from Simulations. Chemical Reviews. 2006; 106:1527. [PubMed: 16608190]
- [16]. Siu SWI, Vacha R, Jungwirth P, Bockmann RA. Molecular simulations of membranes: Physical properties from different force fields. J. Chem. Phys. 2008; 128:125103. [PubMed: 18376978]
- [17]. Holcomb CD, Clancy P, Thompson SM, Zollweg JA. A critical study of simulations of the Lennard-Jones liquid-vapor interface. Fluid Phase Equilibria. 1992; 75:185.
- [18]. Schuler LD, Daura X, van Gunsteren WF. An improved GROMOS96 force field for aliphatic hydrocarbons in the condensed phase. J. Comput. Chem. 1996; 22(11):1205.
- [19]. Marrink SJ, Vries A. H. d. Mark AE. Coarse grained model for semiquantitative lipid simulations. J. Phys. Chem. B. 2004; 108:750.
- [20]. Marrink SJ, Risselada S, Yefimov S, Tieleman DP, Vries A. H. d. The MARTINI forcefield: coarse grained model for biomolecular simulations. J. Phys. Chem. B. 2007; 111:7812. [PubMed: 17569554]

- [21]. Monticelli L, Kandasamy S, Periole X, Larson R, Tieleman DP, Marrink SJ. The MARTINI coarse grained force field: extension to proteins. *J. Chem. Theory Comput.* 2008; 4:819.
- [22]. Srinivas G, Klein ML. Molecular Dynamics simulation of self-assembly and nanotube formation by amphiphilic molecules in aqueous solution: a coarse-grain approach. *Nanotechnology.* 2007; 18:205703.
- [23]. Ensing B, Nielsen SO, Moore PB, Klein ML. Energy conservation in Adaptive Hybrid Atomistic/Coarse-Grain Molecular Dynamics. *J. Chem. Theory Comput.* 2007; 3:1100.
- [24]. Lopez CF, Nielsen SO, Srinivas G, DeGrado WF, Klein ML. Probing Membrane Insertion Activity of Antimicrobial Polymers via Coarse-Grain Molecular Dynamics. *J. Chem. Theory and Comput.* 2006; 2:649. [PubMed: 18985168]
- [25]. Friedrichs MS, Eastman P, Vaidyanathan V, Houston M, Legrand S, Beberg AL, Ensign DL, Burns CM, Pande VS. Accelerating molecular dynamics on graphics processing units. *J. Comput. Chem.* 2009; 30:864–872. [PubMed: 19191337]
- [26]. Yasuda K. Two-Electron Integral Evaluation on the Graphics Processor Unit. *J. Comp. Chem.* 2008; 29:334–342. [PubMed: 17614340]
- [27]. Umfimtsev IS, Martinez TJ. Graphical Processing Units for Quantum Chemistry. *Computing in Science and Engineering* November/December. 2008:26–34.
- [28]. Stone JE, Hardy DJ, Umfimtsev IS, Schulten K. GPU-accelerated molecular modeling coming of age. *J. Mol. Graph. Model.* 2010; 29:116–125. [PubMed: 20675161]
- [29]. Levine B, Stone JE, Kohlmeyer A. Fast Analysis of Molecular Dynamics Trajectories with Graphics Processing Units-Radial Distribution Function Histogramming. *J. Comp. Phys.* Accepted.
- [30]. Stone JE, Kohlmeyer A, Vandivort KL, Schulten K. Immersive molecular visualization and interactive modeling with commodity hardware. *Lecture Notes in Computer Science.* 2010; 6454:382–393. [PubMed: 21297882]
- [31]. Hardy J, Stone JE, Schulten K. Multilevel summation of electrostatic potentials using graphics processing units. *J. Parallel Computing.* 2009; 35:164–177.
- [32]. Stone JE, Phillips JC, Freddolino PL, Hardy DJ, Trabuco LG, Schulten K. Accelerating molecular modeling applications with graphics processors. *J. Comp. Chem.* 2007; 28:2618–2640. [PubMed: 17894371]
- [33]. Voelz VA, Bowman GR, Beauchamp K, Pande VS. Molecular Simulation of ab initio protein folding for a millisecond folder NTL9(1-3). *J. Amer. Chem. Soc.* 2010; 132:1526–1528. [PubMed: 20070076]
- [34]. Elsen, E.; Houston, M.; Vaidyanathan, V.; Darve, E.; Hanrahan, P.; Pande, VS. Poster reception - N-Body simulation on GPUs. *SC*; 2006. p. 188
- [35]. Haque, IS.; Pande, VS. Hard Data on Soft Errors: A Large-Scale Assessment of Real-World Error Rates in GPGPU. *CCGRID*; 2010. p. 691-696.
- [36]. Eastman P, Pande VS. efficient nonbonded interactions for molecular dynamics on a graphics processing unit. *Journal of Computational Chemistry.* 2010; 31(6):1268–1272. [PubMed: 19847780]
- [37]. Anderson JA, Lorenz CD, Travesset A. General purpose molecular dynamics simulations fully implemented on graphics processing units. *J. Comp. Phys.* 2008; 227:5342–539.
- [38]. Anderson JA, Travesset A. Molecular Dynamics on Graphic Processing Units: HOOMD to the Rescue. *Computing in Science and Engineering.* :10.
- [39]. Essmann U, Darden T, Lee H, Perera L, Berkowitz ML, Pederson L. A smooth particle mesh Ewald method. *J. Chem. Phys.* 1995; 103:8577.
- [40]. Darden T, York D, Pederson L. Particle mesh Ewald: An $N \log(N)$ method for Ewald sums in large systems. *J. Chem. Phys.* 1993; 98:10089.
- [41]. Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M. CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations. *J. Comp. Chem.* 1983; 4:187–217.
- [42]. Wu Y, Tepper HL, Voth GA. Flexible simple point-charge water model with improved liquid-state properties. *J. Chem. Phys.* 2006; 124:024503. [PubMed: 16422607]

- [43]. Klauda JB, Brooks BR, Alexander J, MacKerell D, Venable RM, Pastor RW. An ab Initio Study of the Torsional Surface of Alkanes and Its Effect on Molecular Simulations of Alkanes and a DPPC Bilayer. *J. Phys. Chem. B*. 2005; 109:5300–5311. [PubMed: 16863197]
- [44]. Bond S, Leimkuhler B, Laird B. The Nosé-Poincaré Method for Constant Temperature Molecular Dynamics. *J. Comput. Phys.* 1999; 151:114–134.
- [45]. Andersen HC. Molecular Dynamics Simulations at Constant Pressure and/or Temperature. *J. Chem. Phys.* 1980; 72:2384–2393.
- [46]. Payne MC, Teter MP, Allan DC, Arias TA, Joannopoulos JD. Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients. *Rev. Mod. Phys.* 1992; 64:10451097.
- [47]. Stich I, Car R, Parrinello M, Baroni S. Conjugate gradient minimization of the energy functional: A new method for electronic structure calculation. *Phys. Rev. B*. 1989; 39:49975004.
- [48]. Bauer BA, Davis JE, Taufer M, Patel S. Molecular Dynamics Simulations of Aqueous Ions at the Liquid-Vapor Interface Accelerated Using Graphics Processors. *J. Comput. Chem.* 2011; 32:375–385. [PubMed: 20862755]
- [49]. Grand, SL.; Goetz, AW.; Xu, D.; Poole, D.; Walker, RC. 2010. Website, <http://ambermd.org/gpus/>
- [50]. Anderson JA, Lorenz CD, Travesset A. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics*. 2008; 227:5342–5359.
- [51]. Friedrichs M, Eastman P, Vaidyanathan V, Houston M, Grand SL, Beberg A, Ensign D, Bruns C, Pande V. Accelerating molecular dynamic simulation on graphics processing units. *J. Computational Chemistry*. 2009; 30:864872.
- [52]. Harvey M, Giupponi G, Fabritiis GD. ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. *J. Chem. Theory and Comput.* 2009; 5:1632–1639.
- [53]. Allen, MP.; Tildesley, DJ. *Computer Simulation of Liquids*. Clarendon Press; Oxford: 1987.
- [54]. Klauda JB, Ku erka N, Brooks BR, Pastor RW, Nagle JF. Simulation-Based Methods for Interpreting X-Ray Data from Lipid Bilayers. *Biophys. J.* 2006; 90:2796–2807. [PubMed: 16443652]
- [55]. Davis JE, Rahaman O, Patel S. Molecular Dynamics Simulations of a DMPC Bilayer Using Non-Additive Interaction Models. *Biophys. J.* 2009; 96:385–402. [PubMed: 19167291]
- [56]. Czub J, Baginski M. Comparative Molecular Dynamics Study of Lipid Membranes Containing Cholesterol and Ergosterol. *Biophysical Journal*. 2006; 90:2368. [PubMed: 16399829]
- [57]. Pandit SA, Bostick D, Berkowitz ML. Molecular Dynamics Simulation of a Dipalmitoylphosphatidylcholine Bilayer with NaCl. *Biophysical Journal*. 2003; 84:3743–3750. [PubMed: 12770880]
- [58]. Pandit SA, Bostick D, Berkowitz ML. Mixed Bilayer Containing Dipalmitoylphosphatidylcholine and Dipalmitoylphosphatidylserine: Lipid Complexation, Ion Binding, and Electrostatics. *Biophysical Journal*. 2003; 85:3120–3131. [PubMed: 14581212]
- [59]. Marrink SJ, Jahnig F, Berendsen HJ. Proton transport across transient single-file water pores in a lipid membrane studied by molecular dynamics simulations. *Biophysical Journal*. 1996; 71:632–647. [PubMed: 8842203]
- [60]. Ulmschneider MB, Doux JPF, Killian JA, Smith JC, Ulmschneider JP. Mechanism and Kinetics of Peptide Partitioning into Membranes from All-Atom Simulations of Thermostable Peptides. *JACS*. 2010; 132:3452–3460.
- [61]. Wong-ekkabut J, Xu Z, Triampo W, Tang I-M, Tieleman DP, Monticelli L. Effect of Lipid Peroxidation on the Properties of Lipid Bilayers: A Molecular Dynamics Study. *Biophysical Journal*. 2007; 93:4225–4236. [PubMed: 17766354]
- [62]. Ganesan, N.; Bauer, B.; Patel, S.; Taufer, M. FENZI: GPU-enabled Molecular Dynamics Simulations of Large Membrane Regions based on the CHARMM force field and PME; Proceedings of the Tenth IEEE International Workshop on High Performance Computational Biology (HiCOMB); 2011;
- [63]. Harvey, M.; Giupponi, G.; Fabritiis, GD. 2010. Website, <http://www.acellera.com/acemd/performance/>

- [64]. 2010. GROMACS Performance on GPU, Website, <http://www.gromacs.org>
- [65]. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K. Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry*. 2005; 26:1781–1802. [PubMed: 16222654]
- [66]. 2010. NAMD Performance Benchmarks, Website, <http://www.ks.uiuc.edu/Research/namd/performance.html>

Figure 01.
Example of influence region of three charges. The combined effect of the multiple charges impacts the volume under intersection of neighborhoods.

Figure 02.
Example of charge update for which only the affected lattice points are updated.

Figure 03.

When a single cell is affected due to displacement of multiple charges, it is updated with the help of integer atomic intrinsics.

Figure 04.

Visual representations of the lipid-bilayer systems considered in this study. The DMPC 1×1 system describes the smallest system of 72 lipid molecules (36 lipids/leaflet). DMPC 2×2 and 4×4 describe systems with 288 and 1152 lipid molecules, respectively.

Figure 05.

Plot of the total energy fluctuations of DMPC 1x1 as a function of time step size for different time step size - i.e., 0.125 fs, 0.25 fs, 0.5 fs, 1 fs, and 2 fs using single and double precisions as well as two different sets of cutoff values.

Figure 06.

Comparison of the temperature and energy profiles of DMPC 1×1 for 3 ns of NVT MD simulation with 1fs step size for CHARMM on single core, 64 bits and FEN ZI on GTX 480, 32 bits. Shown from left to right, top to bottom are the temperature, and bond, angle, Urey-Bradley, improper torsion, dihedral, Van Der Waals, electrostatic, particle-mesh Ewald energies for CHARMM (black) and FEN ZI (red). All energies are expressed in kcal/mol, and temperature is expressed in Kelvin.

Figure 07.

Comparison of performance of DMPC 1×1 in terms of ns/day for CHARMM on 1, 2, 4, and 8 CPU cores versus FEN ZI on GTX 480 (Fermi chip).

Figure 08.

Simulations of three lipid bilayer membranes (DMPC) with three different sizes, each four time larger than the previous.

(a) Small (DMPC 1x1)

(b) Medium (DMPC 2x2)

(c) Large (DMPC 4x4)

Figure 09.

Number density profiles for various headgroup atoms in small, medium and large DMPC systems. Number densities are expressed in units of $\text{\AA}^{-3}$.

(a) Small (DMPC 1x1)

(b) Medium (DMPC 2x2)

(c) Large (DMPC 4x4)

Figure 10.

Electron density profiles for small, medium and large DMPC membranes. Experimental profile is derived from experimental form factors in Ref.⁵⁴.

Figure 11.

Average orientation of water molecules for small, medium and large DMPC membranes as a function of z position relative to the center of the bilayer (left panels). The right panels show the orientation profiles scaled by a factor, $\zeta = \rho_{\text{fit}}(z)/\rho_{\text{bulk}}$. Here, $\rho_{\text{fit}}(z)$ is taken from an error function fit (Eq. 11) to the z -dependent water density profile. The horizontal line denoting zero is included for clarity in the right panels.

(a) Small (DMPC 1x1)

(b) Medium (DMPC 2x2)

(c) Large (DMPC 4x4)

Figure 12.

Distribution of the angle formed between the $\vec{PN}$ vector and the z -axis for small, medium and large DMPC membranes.

Figure 13.
 $sn - 1$ and $sn - 2$ order parameters for small, medium and large DMPC membranes.

(a) Small (DMPC 1x1)

(b) Medium (DMPC 2x2)

(c) Large (DMPC 4x4)

Figure 14.

Average water dipole moment as a function of the z position relative to the center of the bilayer for small, medium and large DMPC membranes.

(a) Small (DMPC 1x1)

(b) Medium (DMPC 2x2)

(c) Large (DMPC 4x4)

Figure 15. Surface potentials and contributions of water and DMPC to the surface potential for small, medium and large membranes. All potentials are expressed in Volts.

Figure 16.
FEN ZI performance in ns/day of DHFR simulations using different time steps and GPUs.

TABLE I

Surface potential and contributions from water and DMPC for each of the systems studied in this work. All potentials are expressed relative to the potential in the center of the lipid membrane

System	$\Delta\Phi_{\text{Total}}$ (V)	$\Delta\Phi_{\text{Water}}$ (V)	$\Delta\Phi_{\text{DMPC}}$ (V)
Small (DMPC 1×1)	−0.99	−3.42	2.39
Medium (DMPC 2×2)	−1.03	−3.43	2.33
Large (DMPC 4×4)	−1.00	−3.51	2.49