Published in final edited form as:

J Comput Chem. 2018 August 05; 39(21): 1682–1689. doi:10.1002/jcc.25339.

Molecular Dynamics Simulations Using the Drude Polarizable Force Field on GPUs with OpenMM: Implementation, Validation, and Benchmarks

Jing Huang^{1,#}, Justin A. Lemkul^{1,§}, Peter K. Eastman², and Alexander D. MacKerell Jr.^{1,*}

¹Department of Pharmaceutical Sciences, University of Maryland, Baltimore, Baltimore, MD 21201

²Department of Bioengineering, Stanford University, Stanford, CA 94035

Abstract

Presented is the implementation of the Drude force field in the open-source OpenMM simulation package allowing for access to graphical processing unit (GPU) hardware. In the Drude model, electronic degrees of freedom are represented by negatively charged particles attached to their parent atoms via harmonic springs, such that extra computational overhead comes from these additional particles and virtual sites representing lone pairs on electronegative atoms, as well as the associated thermostat and integration algorithms. This leads to an approximately 4-fold increase in computational demand over additive force fields. However, by making the Drude model accessible to consumer-grade desktop GPU hardware it will be possible to perform simulations of one microsecond or more in less than a month, indicating that the barrier to employ polarizable models has largely been removed such that polarizable simulations with the classical Drude model are readily accessible and practical.

Graphical Abstract

Implementation of the polarizable classical Drude force field in the open-source OpenMM simulation package allows for access to graphical processing unit (GPU) hardware making it possible to perform fully polarizable simulations of one microsecond or more in less than a month's time.

*Corresponding Author: Mailing address: 20 Penn St., Room 633, Baltimore, MD, 21201, alex@outerbanks.umaryland.edu, Phone: (410) 706-7442, Fax: (410) 706-5017.

#current address: Westlake University, Hangzhou, Zhejiang, China

§current address: Department of Biochemistry, Virginia Tech, Blacksburg VA 24061

Keywords

graphical processing units; Drude oscillator; polarizable force field; molecular dynamics simulation; CHARMM

INTRODUCTION

Molecular dynamics (MD) simulations are integral in investigating biological, chemical, and physical processes with atomistic resolution, with applications in understanding function, disease, and advancing drug design. The reliability of any MD simulation is dependent upon adequate sampling of the conformational ensemble of the molecule(s) of interest, which in turn relies on an accurate model of the interactions within the system and sufficiently long simulations that can access the relevant states. MD simulations typically apply empirical force field models to compute the energy of a configuration and subsequently the forces acting on all of the particles in the system that are used to propagate motion according to classical mechanics and produce the conformational ensemble. These force fields consist of internal (bonded) terms and nonbonded terms (van der Waals and electrostatic interactions). These calculations are computationally demanding, requiring high-performance software and hardware to carry out simulations of adequate length.

The accuracy of the force field model is a principal concern, and force fields that have existed for decades are continually undergoing revision to improve agreement with theoretical and experimental properties. One critical issue in force field development is the treatment of electronic polarization.¹ Traditional, fixed-charge force fields do not explicitly account for this effect, instead relying on a mean-field approximation for assignment of partial charges to each atom.² That is, multi-body effects are ignored and the total electrostatic interaction energy is computed as the sum of the pairwise interactions in the system. This approximation may fail to capture all of the interactions, particularly in complex environments such as biological systems that are comprised of different

microenvironments. Recently, several force fields have been developed that allow for the explicit representation of electronic polarization in a computationally tractable manner; these include models based on induced dipoles (e.g. AMOEBA),^{3–7} fluctuating charge or charge equilibration models^{8–10} and a force field based on the classical Drude oscillator model.¹¹ As a result of the increased complexity of polarizable simulations (i.e. the increased number of calculations required for the AMOEBA model and the greater number of particles in Drude systems), they are computationally more expensive than their additive counterparts.

Carrying out computationally intensive simulations requires efficient software algorithms and powerful hardware capable of handling the processing and memory requirements of these calculations. The rise of graphical processing units (GPUs) as a platform for MD simulations has greatly enhanced the ability of MD simulations to be carried out on larger systems and extensive time scales. By making use of the considerable computing capacity and fast memory access of the GPU, MD simulations can be performed in a fraction of the time as would be necessary on a parallel supercomputer with standard central processing unit (CPU) hardware. Capitalizing on these powerful tools, GPU implementations of popular MD packages such as CHARMM,¹² AMBER,¹³ NAMD,¹⁴ and GROMACS¹⁵ are available. OpenMM¹⁶ is another such implementation, and its calculations are carried out entirely on the GPU, making it very efficient for performing simulations. It has a flexible application programming interface (API) that can be easily modified through popular programming languages such as Python, allowing new features and energy functions be implemented in a straightforward manner. A CHARMM-OpenMM interface is available and the AMOEBA model is available in OpenMM, and a Tinker-OpenMM interface has been made available for computing free energies using the AMOEBA model in OpenMM.¹⁷

In this work, we present a complete implementation of the Drude polarizable force field model in OpenMM using the CUDA platform, including the recently introduced NBTHOLE^{18,19} and Drude “hard wall”²⁰ features, to allow for fast and stable simulations of polarizable systems. We begin by describing the underlying integration algorithms and features of the Drude model, the changes made to the OpenMM parser for CHARMM force field and topology files, newly added energy functions, and the integration algorithm introduced to carry out the simulations. We then present the validation of our implementation using several biological simulation systems. We conclude with a benchmark of representative biomolecular systems performed on several commonly available GPU platforms.

IMPLEMENTATION DETAILS

The Drude Oscillator Model

A polarizable force field based on the classical Drude oscillator model has recently been reviewed,¹¹ but a brief overview of its important features is appropriate here. Electronic polarization is modeled by attaching auxiliary particles (Drude oscillators) to the real atoms in the system via harmonic springs.²¹ In principle, any atom can be treated as polarizable, but our current Drude force field treats hydrogen atoms as non-polarizable. That is, only non-hydrogen atoms have corresponding Drude oscillators attached to them. Topologically, the Drude oscillators inherit the same pair and exclusion lists as their parent atoms; the bond

between the atom and its Drude oscillator does not result in new lists being propagated and no angle or dihedral interactions involve the Drude oscillators. The Drude oscillators carry negative charge and therefore represent the electronic degrees of freedom in the system. The total charge of a Drude-atom pair, q_{Tot} is defined as:

$$q_{Tot} = q_A + q_D \quad (1)$$

where q_A and q_D are the charges on the parent atom and its Drude oscillator, respectively; q_D is assigned according to the atomic polarizability of the parent atom, α :

$$\alpha = \frac{q_D^2}{k_D} \quad (2)$$

where k_D is the force constant of the Drude-atom harmonic bond.

One important feature of the Drude model is the explicit inclusion of 1–2 and 1–3 interactions involving Drude oscillators. That is, the dipole-dipole interactions between Drude-atom pairs separated by one or two covalent bonds are explicitly included in the energy function, with their Coulombic interactions screened by the following expression proposed by Thole:²²

$$S_{ij}(r_{ij}) = 1 - \left[1 + \frac{ar_{ij}}{2(\alpha_i\alpha_j)^{1/6}} \right] \exp \left[\frac{-ar_{ij}}{(\alpha_i\alpha_j)^{1/6}} \right] \quad (3)$$

where r_{ij} is the distance separating atoms i and j , α_i and α_j are the respective atomic polarizabilities of atoms i and j , and a is the Thole screening parameter determining the extent of screening of the Coulombic potential; the value of a originally defaulted to 2.6 but in the current Drude polarizable force field it is given by $a=a_i+a_j$ where a_i and a_j are atomic Thole parameters that are tuned as part of the force field optimization to more accurately model molecular polarizability.¹¹ As part of the Drude implementation in OpenMM, the Thole screening function has been introduced.

The treatment of electronic polarization is generally isotropic, such that the Drude self-polarization energy, U_{self} is calculated as:

$$U_{self} = \frac{1}{2}k_D\mathbf{d}^2 \quad (4)$$

where $\mathbf{d}$ is the Drude-atom bond length. To better represent the interactions of hydrogen bond acceptors with water and electronegative groups with ions, the self-polarization term

can be described anisotropically by expanding the isotropic force constant k_D into a tensor $\mathbf{K}^{(D)}$, which has off-diagonal elements set to zero. Thus, the self-polarization term becomes:

$$U_{self} = \frac{1}{2}([\mathbf{K}_{11}^{(D)}]d_1^2 + [\mathbf{K}_{22}^{(D)}]d_2^2 + [\mathbf{K}_{33}^{(D)}]d_3^2) \quad (5)$$

where d_1 , d_2 , and d_3 are elements of $\mathbf{d}$ along three axes determined by the local molecular reference frame, as described elsewhere.^{11,23} Further, atoms treated as anisotropic may be augmented with off-site, negatively charged virtual sites (“lone pairs”) that improve the water and ion interactions as a function of orientation as well as the representation of higher-order multipole moments.²³ The OpenMM implementation of the Drude model has full support for anisotropy, and lone pairs are constructed using the existing virtual site functionality in OpenMM (see below).

Updated Force Field Parameter File Parser

Drude force field topologies and parameters are distributed as CHARMM-formatted stream files. They are read as plain text and are called directly in the same manner in OpenMM as any other CHARMM parameter file. For example, to load the parameters for proteins, water, and ions:

```
from simtk.openmm import app
param = app.CharmmParameterSet('toppar_drude_master_protein_2013e.str')
```

The CHARMM parameter parser has been extended to read parameter sections that are specific to the Drude force field such as the NBTHOLE section. NBTHOLE defines the screening between any pair of polarizable atoms that are not bonded, typically involving ions,^{18,19} using the same functional form as the 1–2 and 1–3 dipole screening described above. The difference is that in standard 1–2 and 1–3 electrostatic screening, only dipole-dipole interactions are screened with charge-dipole and charge-charge interactions being fully excluded, whereas in the case of NBTHOLE screening, charge-charge, charge-dipole, and dipole-dipole interactions are all included in the potential energy calculations with the Thole screening function applied.

The NBTHOLE feature has been implemented as a CustomNonbondedForce class in OpenMM. Based on the parameter files that define pairs of atom types to which NBTHOLE screening is applied, particles are added to an interaction matrix that contains all possible screened interactions, along with the atomic and Drude charges and atomic α values. Then, a new class of customized nonbonded interactions $[S_{ij}(r_{ij}) - 1] q_i q_j / r_{ij}$ were added, with $S_{ij}(r_{ij})$ defined in Eq. 4 and the screening factor a being evaluated using the 2DTabulatedFunction in OpenMM. As the $[S_{ij}(r_{ij}) - 1]$ function decays very fast with r , a hard cutoff of 5 Å is applied for NBTHOLE, consistent with the default value used in the CHARMM implementation.

Updated Topology Construction and System Setup

The Drude functional form shares many features with the additive CHARMM force field,²⁴ including treatment of bonded forces, the Lennard-Jones form to treat pairwise van der Waals interactions and corresponding off-diagonal (NBFIX) terms, including terms involving Drude particles,^{25,26} and Coulomb's Law to treat electrostatic interactions between point charges. Additional elements of the functional form include explicit inclusion of screened 1–2 and 1–3 dipole-dipole interactions and anisotropic polarization. These elements of the functional form are implemented in a DrudeForce class, which contains a list of all Drude oscillators, the atoms to which they are connected, a list of atoms treated with anisotropic polarization [Eq. (5)], and all the 1–2 and 1–3 Thole screened dipole pairs. The topology of the system is read from a CHARMM PSF file by the CharmmPsfFile class in OpenMM. The PSF format for a Drude polarizable system is slightly different from a non-polarizable system, as it contains more information in the description of the atoms as well as the inclusion of new sections in the PSF related to anisotropy and lone pairs. The CharmmPsfFile class has been enhanced to parse PSF files for both non-polarizable and polarizable systems.

To populate the DrudeForce class, the atoms in the PSF file are read, including atomic polarizability (α_i) and atomic Thole factor (a_i). From the bonded connectivity of the molecules specified in the PSF file, a list of 1–2 and 1–3 neighbors is generated, specifying the list of Drude-atom pairs to which Thole screening will be applied via a function called addScreenedPair, which stores a list of charges, atomic polarizabilities, and values of the Thole screening factor $a = a_i + a_j$. A list of anisotropic polarization entries is also stored in DrudeForce. This list includes the Drude oscillator, the atom to which it is connected, three additional particles that define the local molecular coordinate system, and elements of the force constant tensor. The anisotropy entries are read from the PSF under the heading !NUMANISO, which contains a list of the force constants for Eq. (5), followed by sets of four constructing atom numbers for the local geometry. The Drude oscillator is identified as being bonded to the first atom in this each set of four atoms.

Lone pairs are off-center virtual sites that have their positions constructed from parent atoms based on specified geometric parameters. As with any virtual site, during dynamics, any forces acting on the lone pairs are then propagated onto those constructing atoms. The lone pairs thus do not have their positions updated like other dynamical atoms; instead, they are constructed after the position update for a given MD time step. Within the list of atoms in the PSF file, a lone pair is identified by having a particle type of –1, while all other atoms are assigned a particle type of 0. The constructing parameters for the lone pairs are listed under the !NUMLP heading in the PSF file. As with the anisotropy entries, the parameters for each lone pair construction are listed, followed by sets of four atoms that are used as reference positions for constructing the lone pairs as atomic positions are updated during dynamics.

In concert with the interpreted PSF file, the user must supply coordinates for the initial configuration of the simulation, which can be either in Protein Databank (PDB) format or in CHARMM CRD format. These files are read by the PDBFile or CharmmCrdFile classes in OpenMM.

Integration Algorithm

The integration of the equations of motion is carried out via the velocity Verlet method, the same as the original implementation in CHARMM²⁷ and subsequent implementation in GROMACS,²⁸ and differing from the Drude implementation in NAMD, which employs Langevin dynamics.²⁹ There are two methods for propagating the positions of the Drude oscillators over time, both based on the Born-Oppenheimer approximation that electronic degrees of freedom relax instantaneously on the time scale of the motion of parent nuclei. The first method relies on a self-consistent field (SCF) relaxation of the Drude oscillators at each time step. That is, for a fixed configuration of the atoms, the Drude oscillators in the system are energy-minimized until the gradient with respect to the Drude oscillator positions is zero:

$$\frac{\partial U}{\partial \mathbf{d}_i} = 0 \quad (6)$$

where U is the potential energy of the system and $\mathbf{d}_i$ is the position of the Drude oscillators. In practice, achieving a gradient of exactly zero is difficult, if not impossible, given limits in machine precision, and instead, a maximum allowable force is specified to indicate the tightness of the SCF convergence. This value is typically on the order of 0.1 kJ mol⁻¹ nm⁻¹. The SCF integration scheme is implemented in OpenMM as the DrudeSCFIntegrator and is called as follows:

```
from simtk import openmm
integrator = openmm.DrudeSCFIntegrator(0.002*picoseconds)
integrator.setMinimizationErrorTolerance(0.1)
```

The drawback of the SCF approach is that it is computationally expensive, requiring multiple force evaluations per MD time step. Therefore, more efficient methods are preferable. Lamoureux and Roux devised an extended Lagrangian integration scheme for polarizable systems.²⁷ In this approach, the positions of the Drude oscillators are integrated on equal footing as normal atoms and their dynamics are coupled to their parent atoms. To accomplish this integration, the system is coupled to two thermostats, one “physical” thermostat responsible for scaling the velocities of the normal atoms at the desired system temperature, and one thermostat that maintains the relative motion of the Drude oscillators at a very low temperature, typically 1 K. This approach was shown to approximate the SCF surface.²⁷ The extended Lagrangian approach was initially conceived with dual Nosé-Hoover thermostats^{27,30,31} to scale velocities, but recent implementations in CHARMM and NAMD also allow for Langevin damping.^{12,29} It is the latter approach that is implemented in OpenMM.

To initiate an extended Lagrangian dynamics simulation with the velocity Verlet integrator using a 1-fs time step and maintaining the temperature of the “physical” thermostat at 298 K and the relative Drude thermostat at 1 K, with thermostat friction coefficients of 5 ps⁻¹ and 20 ps⁻¹, respectively, the user would call:


```

from simtk import openmm
integrator = openmm.DrudeLangevinIntegrator(298*kelvin, 5/picosecond,
1*kelvin, 20/picosecond, 0.001*picoseconds)

```

We note that with the extended Lagrangian integrator, the integration time step is typically limited to 1 fs while a 2-fs time step may be used with SCF-based integration, though the former is still significantly less computationally demanding than the latter.

“Hard Wall” Constraint

The fastest motions in a simulated system limit the time step that can be applied. For non-polarizable systems, these motions are typically bond vibrations involving hydrogen atoms. For polarizable systems, the Drude-atom bonds oscillate at the highest frequencies, which sets the limit for the maximum integration time step. A problem facing polarizable systems is so-called “polarization catastrophe,” in which forces build up on a Drude oscillator and lead to large Drude displacements and instability. Early polarizable simulations were limited to very short time steps (~0.5 fs) to circumvent this problem, rendering them inefficient. In time, approaches such as the application of a quartic restraining potential were implemented, allowing for the use of a 1-fs time step,¹⁹ which, while it was an improvement, remained prone to instability. To solve this issue, Chowdhary et al. introduced a “hard wall” constraint that reflects any Drude oscillator back towards its parent atom along the Drude-atom bond if its displacement reaches a user-defined limit, typically 0.2 or 0.25 Å.²⁰ By scaling the velocities of the Drude-atom pairs to which the hard wall is applied, momentum is conserved. Using the hard wall constraint in MD simulations, in concert with normal constraints on bonds involving hydrogen, allows the maximum time step to be set to 1 fs, reasonable for carrying out simulations efficiently, as presented below.

In OpenMM, after the integrator has been initialized (see above), the Drude hard wall constraint can be added as follows:

```

integrator.setMaxDrudeDistance(0.02)

```

The above call sets the maximum allowable Drude-atom bond length at 0.02 nm (0.2 Å), the value recommended for use in polarizable simulations with the Drude model.²⁰ Note that the hard wall constraint requires the use of DrudeLangevinIntegrator, as it integrates the positions of the Drude oscillators, which are coupled to a relative thermostat and therefore possess velocities. In the SCF approach, the Drude oscillators have zero velocity at all time steps. To avoid polarization catastrophe when using the DrudeSCFIntegrator, an anharmonic restraining potential between the Drude oscillators and their parent atoms can be invoked using the OpenMM CustomBondForce class as follows:

```

psf=openmm.CharmmPsfFile('system.psf')
hyper=openmm.CustomBondForce('step(r-rhyper)*((r-rhyper)*khyper)^powh')
hyper.addGlobalParameter('khyper', 100.0)

```



```

hyper.addGlobalParameter('rhyper', 0.02)
hyper.addGlobalParameter('powh', 6)
system.addForce(hyper)
for d in psf.drudepair_list:
    hyper.addBond(d[0], d[1], [])

```

In this restraining potential, the user is free to specify the exponent used (powh); typical values are 4 or 6.

VALIDATION AND BENCHMARKING SIMULATIONS

Validation of the Dual-Langevin integrator in OpenMM

We have followed the procedure proposed by Shirts³² to validate that the correct statistical ensembles can be achieved using the Drude dual-Langevin integrator implemented in OpenMM. Briefly, if one pair of canonical (NVT) simulations was carried out with two different temperatures, T_1 and T_2 , the distribution P of potential energies E should satisfy the following linear relationship with respect to E :

$$\ln \frac{P(E | \beta_2)}{P(E | \beta_1)} = [\beta_2 A_2 - \beta_1 A_1] + [\beta_1 - \beta_2] E \quad (7)$$

where $\beta_1 = (k_B T_1)^{-1}$ and $\beta_2 = (k_B T_2)^{-1}$ are inverse temperatures, and A_1 and A_2 are Helmholtz free energies. While the intercept of Eq. 7 is generally unknown, the linearity itself, and its slope α satisfies $\alpha = \beta_1 - \beta_2$, providing a useful tool to examine ensemble consistency.

Five NVT simulations of 1024 SWM4-NDP water molecules were carried out with temperatures spanning from 300 K to 320 K with an interval of 5 K, resulting in four pairs of simulations to be examined. The water molecules were placed in a cubic box of 38.6 Å, with full periodic boundary conditions applied. The particle mesh Ewald (PME) method^{33,34} was used to calculate electrostatic interactions, with a real-space cutoff of 12 Å. The van der Waals potential was switched to zero from 10 – 12 Å. Bonds to hydrogen were constrained and the hard wall constraint was set for all Drude-atom bonds, with a limit of 0.2 Å. The reference temperature of the “physical” thermostat was maintained at the desired temperature with a friction coefficient of 5 ps⁻¹. The relative Drude thermostat was set to 1 K with a friction coefficient of 20 ps⁻¹. Each simulation was carried out for 1 μs with a time step of 1 fs, and potential energies were evaluated every 1 ps, resulting a total of 10⁶ potential energies to be assembled into histograms, with a bin size of 100 kJ mol⁻¹.

As shown in Fig. 1, a linear relationship was observed for all four pairs of simulations examined. The slopes evaluated by fitting the simulation distributions to Eq. 7 are also consistent with theoretical values. For example, a slope of 0.00653 mol kJ⁻¹ was obtained by comparing the 300 K and the 305 K simulations, which matches the theoretical value of 0.00657 mol kJ⁻¹ with a relative deviation of 0.6%. The largest relative deviation found is 1.7%, from taking the ratio between the 305 K and 310 K potential energy distributions. As

a comparison, the Berendsen thermostat, still sometimes used despite being known for its inability to produce a correct canonical distribution of velocities, was found to yield a relative deviation of 140% when simulating a box of 900 TIP3P water molecules.³²

Comparison of forces generated by CHARMM and OpenMM

As described above, the classical Drude polarizable force field introduces a number of additional potential energy terms compared to the conventional fixed-charge force fields. It is thus important to validate that the potential energy function has been correctly implemented, as well as its first derivatives that are used to propagate systems in MD simulations. To this end, we compared the forces of three different condensed phase simulation systems generated by CHARMM and OpenMM. We evaluated protein (ubiquitin, PDB 1UBQ)³⁵, DNA (Dickerson-Drew dodecamer, EcoRI, PDB 1BNA),³⁶ and 1 M MgCl₂ systems, all taken from previous studies.^{25,37,38} Each system was solvated in SWM4-NDP water³⁹ and the EcoRI system included ~100 mM NaCl. The differences between forces computed by the two programs were evaluated for each atom in the systems using the relative errors defined by the L2 norm between its (x,y,z) component:

$$e = \sqrt{\sum_{i=x,y,z} \left(\frac{F_i^{\text{OpenMM}} - F_i^{\text{CHARMM}}}{F_i^{\text{CHARMM}}} \right)^2} \quad (8)$$

Ten different configurations were considered for each system using snapshots taken from previous MD simulations, and results were pooled together. As shown in the Table 1, the mean and the median errors are on the order of 10⁻⁴. Also reported are the 90th percentile and the 99th percentile, which indicate that the relative force errors that 90 percent of atoms and 99 percent of atoms have are all less than 10⁻³. We note that different MD engines use slightly different Coulombic constants, so in order to eliminate its impact on the force comparison, the same 1/4πε value (332.063709 kcal mol⁻¹ Å⁻¹) has been used for the CHARMM and OpenMM calculation results presented in Table 1. If we apply the values used in the respective programs (332.0716 kcal mol⁻¹ Å⁻¹ for CHARMM and 332.063709 kcal mol⁻¹ Å⁻¹ for OpenMM) to evaluate the forces, a marginal impact on the errors is observed. The mean relative errors in forces change from 2.0×10⁻⁴, 1.9×10⁻⁴ and 3.7×10⁻⁴ to 2.0×10⁻⁴, 2.0×10⁻⁴ and 3.7×10⁻⁴ for the three systems considered, and the 99th percentiles increase slightly from 8.2×10⁻⁴, 7.1×10⁻⁴ and 9.9×10⁻⁴ to 8.3×10⁻⁴, 7.2×10⁻⁴ and 9.9×10⁻⁴, respectively.

Benchmark Simulations

Benchmarking simulations were carried out on the three systems used in the force comparison. An additional system containing a DNA 24mer in 100 mM NaBr in a very large simulation box of nearly 650,000 total particles was constructed to act as an extreme test of OpenMM performance for polarizable systems. Simulations were carried out using the DrudeLangevinIntegrator implemented in OpenMM with a time step of 1 fs, and periodic boundary conditions were applied. The nonbonded interactions were treated the same as in the SWM4-NDP simulations described above. Bonds to hydrogen were constrained and the

hard wall constraint was set for all Drude-atom bonds, with a limit of 0.2 Å. Simulations were carried out under an NPT ensemble. The reference temperature of the “physical” thermostat was set to 298 K and maintained with a friction coefficient of 5 ps⁻¹. The relative Drude thermostat was set to 1 K with a friction coefficient of 20 ps⁻¹. Pressure was maintained at 1 bar using the Monte Carlo barostat in OpenMM. A full example input file for running the EcoRI system is provided in the Supporting Information.

Simulations were carried out on a range of commonly available GPU hardware. Specifications of these computing nodes and the resulting performance in ns/day are given in Table 2. Even with a 1-fs integration time step typically used in polarizable MD systems, the simulations are highly efficient, such that on consumer-grade hardware like the NVIDIA GTX 1080Ti, obtaining a microsecond of simulation is achievable in less than a month for all systems except the large DNA system. The moderately sized systems, on the order of tens of thousands of particles, reflect common sizes for most simulations of biomolecules. The speed is roughly 4 times slower than corresponding additive systems where a 2-fs time step is used; for example, simulations of the ubiquitin using the additive C36m FF perform at 83 ns/day on a GTX980 card. In addition, with additive force fields it is possible to achieve longer integration time steps (e.g. 4 fs) through hydrogen mass partitioning, an approach that is not accessible to the Drude model due to the fastest degree of freedom being the Drude oscillator motion.^{40–42} Simulation speeds on the less expensive NVIDIA GTX 980 are approximately 2–2.5 times slower than the GTX 1080Ti, but still yield adequate performance in each case. We note that the performance of the large DNA system on the GTX 980 hardware is hampered by I/O burden; writing frames at 50-ps intervals requires intense use of the CPU that degrades performance. Without writing any frames to the trajectory, the simulation speed is on the order of 1.2 ns/day, but performing a simulation in this manner is generally not useful, as no ensemble of configurations would be obtained. Therefore, a more powerful CPU with more memory is required for performing simulations of this size. The GTX 1080Ti, however, performs well (2.3 ns/day) even for this computationally expensive simulation. Simulation speeds were slowest on the NVIDIA K80 for all systems except the large DNA, generally 1.5–2.5 times slower than the NVIDIA GTX 980. Nevertheless, any of these GPU cards can be used for reasonably efficient polarizable simulations.

CONCLUSIONS

We have presented the implementation of the Drude polarizable force field¹¹ in OpenMM,¹⁶ a high-performance GPU-based simulation package. The robustness of the implementation was evaluated against forces collected from simulations carried out with CHARMM,¹² in which the classical Drude oscillator model was first developed.²⁷ The OpenMM results agree well with the reference CHARMM properties, thus validating the implementation. We have also carried out ensemble consistency tests, which validate the Drude dual-Langevin algorithm itself and its implementation in the OpenMM package. Performance of the code was assessed on three different hardware platforms using CUDA libraries on NVIDIA hardware (GTX 980, GTX 1080Ti, and K80). The GTX 1080Ti performed the best, with simulation speeds that make obtaining microsecond trajectories feasible within a matter of weeks. In addition, the classical Drude model may be used with enhanced sampling

techniques⁴³ and in free energy perturbation calculations,¹⁹ though these capabilities have yet to be implemented in OpenMM. Our current implementation also lays the foundation for implementing and using the quantum Drude model^{44,45} in OpenMM. The OpenMM implementation of the classical Drude model has been uploaded to the public GitHub repository of OpenMM and is therefore freely available for use by the scientific community. The availability of this high-performance implementation of the Drude force field and associated integration algorithms in OpenMM makes running polarizable simulations readily practical and accessible on widely available commodity hardware.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

Financial support for this work was provided by the National Institutes of Health grants GM072558, GM070855, GM051501 (to A.D.M.) and F32GM109632 (to J.A.L.). Computing time on the Comet supercomputer was provided by the National Science Foundation XSEDE project and additional computing resources were provided by the University of Maryland, Baltimore Computer-Aided Drug Design Center.

References

1. Vanommeslaeghe K, MacKerell AD Jr. *Biochim Biophys Acta*. 2015; 1850:861–871. [PubMed: 25149274]
2. MacKerell AD Jr. *J Comput Chem*. 2004; 25(13):1584–1604. [PubMed: 15264253]
3. Ponder JW, Wu C, Ren P, Pande VS, Chodera JD, Schnieders MJ, Haque I, Mobley DL, Lambrecht DS, DiStasio RA Jr, Head-Gordon M, Clark GNI, Johnson ME, Head-Gordon T. *J Phys Chem B*. 2010; 114(8):2549–2564. [PubMed: 20136072]
4. Shi Y, Xia Z, Zhang J, Best R, Wu C, Ponder JW, Ren P. *J Chem Theory Comput*. 2013; 9:4046–4063. [PubMed: 24163642]
5. Cieplak P, Dupradeau F-Y, Duan Y, Wang J. *J Phys: Condens Matter*. 2009; 21(33):333102. [PubMed: 21828594]
6. Kaminski GA, Ponomarev SY, Liu AB. *J Chem Theory Comput*. 2009; 5(11):2935–2943. [PubMed: 20209038]
7. Sharma I, Kaminski GA. *J Comput Chem*. 2017; 38(2):65–80. [PubMed: 27785788]
8. Patel S, Brooks CL III. *J Comput Chem*. 2004; 25:1–16. [PubMed: 14634989]
9. Patel S, MacKerell AD Jr, Brooks CL III. *J Comput Chem*. 2004; 25:1504–1514. [PubMed: 15224394]
10. Banks JL, Kaminski GA, Zhou R, Mainz DT, Berne BJ, Friesner RA. *J Chem Phys*. 1999; 110:741–754.
11. Lemkul JA, Huang J, Roux B, MacKerell AD Jr. *Chem Rev*. 2016; 116(9):4983–5013. [PubMed: 26815602]
12. Brooks BR, Brooks CL III, MacKerell AD Jr, Nilsson L, Petrella RJ, Roux B, Wom Y, Archontis G, Bartels C, Boresch S, Caflisch A, Caves L, Cui Q, Dinner AR, Feig M, Fischer S, Gao J, Hodoseck M, Im W, Kuczera K, Lazaridis T, Ma J, Ovchinnikov V, Paci E, Pastor RW, Post CB, Pu JZ, Schaefer M, Tidor B, Venable RM, Woodcock HL, Wu X, Yan W, York DM, Karplus M. *J Comput Chem*. 2009; 30:1545–1614. [PubMed: 19444816]
13. Case, DA., Cerutti, DS., Cheatham, TE., III, Darden, TA., Duke, RE., Giese, TJ., Gohlke, H., Goetz, AW., Greene, D., Homeyer, N., Izadi, S., Kovalenko, A., Lee, TS., LeGrand, S., Li, P., Lin, C., Liu, J., Luchko, T., Luo, R., Mermelstein, D., Merz, KM., Monard, G., Nguyen, H., Omelyan, I., Onufriev, A., Pan, F., Qi, R., Roe, DR., Roitberg, A., Sagui, C., Simmerling, CL., Botello-

- Smith, WM., Swails, J., Walker, RC., Wang, J., Wolf, RM., Wu, X., Xiao, L., York, DM., Kollman, PA. University of California; San Francisco: 2017.
14. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K. *J Comput Chem*. 2005; 26:1781–1802. [PubMed: 16222654]
15. Abraham M, Murtola T, Schulz R, Páll S, Smith JC, Hess B, Lindahl E. *SoftwareX*. 2015; 1–2:19–25.
16. Eastman P, Friedrichs MS, Chodera JD, Radmer RJ, Bruns CM, Ku JP, Beauchamp KA, Lane TJ, Wang L-P, Shukla D, Tye T, Houston M, Stich T, Klein C, Shirts MR, Pande VS. *J Chem Theory Comput*. 2013; 9(1):461–469. [PubMed: 23316124]
17. Harger M, Li D, Wang Z, Dalby K, Lagardère L, Piquemal J-P, Ponder J, Ren P. *J Comput Chem*. 2017; In Press. doi: 10.1002/jcc.24853
18. Luo Y, Jiang W, Yu H, MacKerell AD Jr, Roux B. *Faraday Discuss*. 2013; 160:135–149. [PubMed: 23795497]
19. Yu H, Whitfield TW, Harder E, Lamoureux G, Vorobyov I, Anisimov VM, MacKerell AD Jr, Roux B. *J Chem Theory Comput*. 2010; 6:774–786. [PubMed: 20300554]
20. Chowdhary J, Harder E, Lopes PEM, Huang L, MacKerell AD Jr, Roux B. *J Phys Chem B*. 2013; 117:9142–9160. [PubMed: 23841725]
21. Drude, P., Millikan, RA., Mann, RC. *The Theory of Optics*. Longmans, Green, and Co; New York: 1902.
22. Thole BT. *Chem Phys*. 1981; 59(3):341–350.
23. Harder E, Anisimov VM, Vorobyov IV, Lopes PEM, Noskov SY, MacKerell AD Jr, Roux B. *J Chem Theory Comput*. 2006; 2:1587–1597. [PubMed: 26627029]
24. MacKerell AD Jr, Bashford D, Bellott M, Dunbrack RL Jr, Evanseck JD, Field MJ, Fischer S, Gao J, Guo H, Ha S, Joseph-McCarthy D, Kuchnir L, Kuczera K, Lau FTK, Mattos C, Michnick S, Ngo T, Nguyen DT, Prodhom B, Reiher WE III, Roux B, Schlenkrich M, Smith JC, Stote R, Straub J, Watanabe M, Wiórkiewicz-Kuczera J, Yin D, Karplus M. *J Phys Chem B*. 1998; 102(18): 3586–3616. [PubMed: 24889800]
25. Lemkul JA, MacKerell AD Jr. *J Phys Chem B*. 2016; 120:11436–11448. [PubMed: 27759379]
26. Lin F-Y, MacKerell AD Jr. *J Chem Theory Comput*. 2018 Submitted.
27. Lamoureux G, Roux B. *J Chem Phys*. 2003; 119:3025–3039.
28. Lemkul JA, Roux B, van der Spoel D, MacKerell AD Jr. *J Comput Chem*. 2015; 36(19):1473–1479. [PubMed: 25962472]
29. Jiang W, Hardy DJ, Phillips JC, MacKerell AD Jr, Schulten K, Roux B. *Journal of Physical Chemistry Letters*. 2011; 2:87–92. [PubMed: 21572567]
30. Nosé S. *J Chem Phys*. 1984; 81(1):511–519.
31. Hoover WG. *Phys Rev A: At Mol Opt Phys*. 1985; 31(3):1695–1697.
32. Shirts MR. *J Chem Theory Comput*. 2013; 9(2):909–926. [PubMed: 26588735]
33. Darden T, York D, Pedersen L. *J Chem Phys*. 1993; 98(12):10089–10092.
34. Essmann U, Perera L, Berkowitz ML, Darden T, Lee H, Pedersen LG. *J Chem Phys*. 1995; 103(19):8577–8593.
35. Vijay-Kumar S, Bugg CE, Cook WJ. *J Mol Biol*. 1987; 194(3):531–544. [PubMed: 3041007]
36. Drew HR, Wing RM, Takano T, Broka C, Tanaka S, Itakura K, Dickerson RE. *Proc Natl Acad Sci USA*. 1981; 78:2179–2183. [PubMed: 6941276]
37. Lemkul JA, MacKerell AD Jr. *J Chem Theory Comput*. 2017; 13(5):2072–2085. [PubMed: 28398748]
38. Lopes PEM, Huang J, Shim J, Luo Y, Li H, Roux B, MacKerell AD Jr. *J Chem Theory Comput*. 2013; 9:5430–5449. [PubMed: 24459460]
39. Lamoureux G, Harder E, Vorobyov IV, Roux B, MacKerell AD Jr. *Chem Phys Lett*. 2006; 418:245–249.
40. Jacucci, G., Rahman, A. Orsay, France: 1974. p. 32-40.
41. Feenstra KA, Hess B, Berendsen HJC. *J Comput Chem*. 1999; 20(8):786–798.

42. Hopkins CW, Le Grand S, Walker RC, Roitberg A. *J Chem Theory Comput.* 2015; 11(4):1864–1874. [PubMed: 26574392]
43. Huang J, MacKerell AD Jr. *Biophys J.* 2014; 107(4):991–997. [PubMed: 25140435]
44. Jones AP, Crain J, Sokhan VP, Whitfield TW, Martyna GJ. *Physical Review B.* 2013; 87(14):144103.
45. Whitfield TW, Martyna G. *J Chemical Physics Letters.* 2006; 424(4):409–413.

Summary

Next generation empirical force fields will include the explicit treatment of electronic polarization. In this study the OpenMM toolkit is extended to treat the classical Drude polarizable force field allowing for simulations on inexpensive GPU hardware. This capability will allow for routine simulation studies of condensed phase macromolecular systems at microsecond time scales thereby facilitating the use of polarizable force fields in computational chemistry and biophysics.

Figure 1.

The logarithm of the ratio of potential energy distributions under two different temperatures follows a linear relationship with respect to the potential energy, E , sampled from five 1 μ s NVT simulations of 1024 Drude polarizable water molecules under 300, 305, 310, 315, and 320 K using the dual-Langevin integrator implemented in OpenMM. The slopes by fitting to a linear equation are consistent with the theoretical values given by $(k_B T_1)^{-1} - (k_B T_2)^{-1}$. Both values are given in the plots with the unit of mol kJ⁻¹. Uncertainties were estimated with block analysis using 5 blocks.

Table 1

The relative errors in forces computed with OpenMM and CHARMM for all atoms in three simulation systems.

System	mean	median	90 th percentile	99 th percentile
1 M MgCl ₂ solution	2.0×10^{-4}	1.6×10^{-4}	3.4×10^{-4}	8.2×10^{-4}
Ubiquitin + water	1.9×10^{-4}	1.6×10^{-4}	3.5×10^{-4}	7.1×10^{-4}
EcoRI + water + 100 mM NaCl	3.7×10^{-4}	1.6×10^{-4}	3.9×10^{-4}	9.9×10^{-4}

Table 2

Benchmarking systems and performance (ns/day) on different GPU hardware.

System	# Atoms	# Particles ^d	NVIDIA GTX 980 ^a	NVIDIA GTX 1080Ti ^b	NVIDIA K80 ^c
1 M MgCl ₂ solution	2,988	4,916	84	226	57.5
Ubiquitin + water	20,292	33,837	22.4	47.4	10.5
EcoRI + water + 100 mM NaCl	24,515	41,015	19	40	8.3
DNA 24mer + water + 100 mM NaBr	388,793	648,419	0.3	2.3	0.5

^aAMD Athlon X2 dual core 4600+ CPU (2.4 GHz), 2 GB CPU memory, CUDA 8.0.44

^bAMD Ryzen 7 1700 8-core CPU (3 GHz), 16 GB CPU memory, CUDA 8.0.44

^cIntel Xeon E5-2680v3 24-core CPU (2.5 GHz), 128 GB CPU memory, CUDA 8.0

^dThe total number of particles includes atoms, Drude oscillators, and lone pairs.

^ePerformance of this system on the GTX 980 is limited by the I/O burden of periodically writing to the trajectory and is therefore not a true indication of the simulation speed as the CPU becomes a limiting factor.