Published in final edited form as:

J Comput Chem. 2015 July 15; 36(19): 1473–1479. doi:10.1002/jcc.23937.

Implementation of Extended Lagrangian Dynamics in GROMACS for Polarizable Simulations Using the Classical Drude Oscillator Model

Justin A. Lemkul¹, Benoît Roux², David van der Spoel³, and Alexander D. MacKerell Jr.^{1,*}

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

²Department of Biochemistry and Molecular Biology, University of Chicago, Chicago, Illinois 60637, USA

³Department of Cell and Molecular Biology, Uppsala University, Uppsala, Sweden

Abstract

Explicit treatment of electronic polarization in empirical force fields used for molecular dynamics simulations represents an important advancement in simulation methodology. A straight forward means of treating electronic polarization in these simulations is the inclusion of Drude oscillators, which are auxiliary, charge-carrying particles bonded to the cores of atoms in the system. The additional degrees of freedom make these simulations more computationally expensive relative to simulations using traditional fixed-charge (additive) force fields. Thus, efficient tools are needed for conducting these simulations. Here, we present the implementation of highly scalable algorithms in the GROMACS simulation package that allow for the simulation of polarizable systems using extended Lagrangian dynamics with a dual Nosé-Hoover thermostat as well as simulations using a full self-consistent field treatment of polarization. The performance of systems of varying size is evaluated, showing that the present code parallelizes efficiently and is the fastest implementation of the extended Lagrangian methods currently available for simulations using the Drude polarizable force field.

Abstract

Polarizable force fields represent the new generation in biomolecular simulation. In the Drude oscillator model, additional particles are attached to atoms in the system to represent electronic degrees of freedom. These simulations are more expensive than those done with traditional force fields, and to this end the powerful GROMACS software package has been extended to include algorithms necessary to efficiently simulate polarizable systems, enabling long-scale simulations of increasingly informative and accurate biomolecular models.

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

AUTHOR CONTRIBUTIONS

J.A.L., B.R., and A.D.M. conceived the project, J.A.L. and D.v.d.S. planned the code implementation, J.A.L. wrote the code and performed benchmarking simulations, J.A.L., B.R., D.v.d.S., and A.D.M. wrote the paper.

Keywords

molecular dynamics; induced polarization; scalability; parallel performance

INTRODUCTION

Most (bio-)molecular dynamics (MD) simulations are carried out using additive (non-polarizable) force fields, which feature atom-centered, fixed partial charges on all of the atoms in the system. These models have been parametrized to respond in an average way to different chemical environments, but they cannot account for multi-body effects or respond to changes in the local electric field. A new generation of force fields has emerged that features explicit inclusion of electronic polarization. Several approaches to treating induced polarization have been developed, including the induced dipole model,^[1–7] the fluctuating charge model,^[8–12] and the classical Drude oscillator model^[13] (also called the “shell model”^[14] or “charge on a spring”).^[15]

In the classical Drude oscillator model, auxiliary particles representing electronic degrees of freedom (“Drude particles”) are attached to their parent atoms via a harmonic spring. The charges on each Drude particle, q_D , and its parent atom, q_A , are calculated from the atomic polarizability, α :

$$\alpha = \frac{q_D^2}{k_D} \quad (1)$$

where k_D is the force constant of the spring, and q_A is calculated as $q_{Tot} - q_D$, where q_{Tot} is the total charge on the Drude-atom pair. In principle, any or all of the atoms can be treated as polarizable. Recently, the Drude-2013 polarizable force field for water,^[16,17] proteins,^[18] DNA,^[19,20] monosaccharides,^[21] and dipalmitoylphosphatidylcholine^[22] has been introduced that treats only non-hydrogen atoms as polarizable. This force field supports

atom-based anisotropic polarization and lone pairs (charge-carrying virtual sites) on hydrogen bond acceptors to improve hydrogen-bonding interactions as a function of orientation.^[23]

The ability of the Drude polarizable force field to allow the molecules being simulated to respond to changes in the local electric field has been shown to have important physical consequences. Studies have shown that the Drude-2013 force field captures the cooperativity of folding of the model (AAQAA)₃ peptide due to the ability of the peptide bond to polarize^[24] and variations in glutamine side-chain dipole moments as a function of χ^1 rotation in ubiquitin have been observed^[25] along with variability of the dipole moments in other amino acids in MD simulations. Concerning DNA, variations in base dipole moments and solvating waters during base flipping^[26] produced near-quantitative agreement with experimental equilibrium constants for base opening. The Drude DNA model has been shown to better model ion-DNA interactions than additive force fields, in addition to predicting that the structure of DNA in solution is sensitive to the type of monocation, a phenomenon not observed in additive force fields.^[20,27] In all of these studies, the Drude-2013 force field produced new insights that cannot be obtained with additive models. It is clear that these types of simulations represent an important development in the field of theoretical (bio)chemistry and biophysics. Since there is an additional computational expense due to the greater number of particles in the system, such simulations will only become commonplace when efficient algorithms are widely available.

Simulations of polarizable systems using this model in the CHARMM^[28] and NAMD^[29] packages have been described previously.^[30,31] In the present work, the Drude-2013 force field has been implemented in the GROMACS simulation package.^[32–34] GROMACS is a highly efficient MD simulation program that scales well over a large range of computing cores. Additionally, it can support MD simulations with graphical processing units (GPU) to further enhance throughput. A modification of the existing velocity Verlet integrator^[35] implemented in GROMACS^[34] is described, as well as new functions required by the force field. The inclusion of the Drude-2013 force field and associated algorithms in GROMACS allows for efficient simulations of polarizable systems.

THEORY AND ALGORITHM

Simulations of Drude polarizable systems require that the positions of the Drude particles be updated in the presence of the electric field generated by the configuration of the system. There are two approaches to this problem. The first uses a self-consistent field (SCF) approach, which has previously been implemented in GROMACS.^[14] The SCF condition follows naturally from the Born-Oppenheimer approximation, in which electronic degrees of freedom are assumed to relax instantaneously in response to the configuration of the atomic nuclei. The SCF condition implies solving the position of the Drude particles ($\mathbf{d}_i$) in the field of the fixed nuclei via energy minimization:

$$\frac{\partial U}{\partial \mathbf{d}_i} = 0 \quad (2)$$

where U is the total potential energy of the system (including the harmonic spring linking the Drude particle to its parent nucleus). In GROMACS, this procedure is carried out at every MD time step, before the positions of the real atoms are updated. An efficient energy-minimization algorithm has been introduced for this process,^[14] after which the normal force routines are called for the remaining atoms. The user sets a maximum allowable tolerance for the gradient in Eq. 2 (typically $1.0 \text{ kJ mol}^{-1} \text{ nm}^{-1}$, as minimization to exactly zero is difficult, if not impossible, with limited precision) during dynamics, as well as a maximum number of iterations of energy minimization per time step. Though theoretically and practically straightforward, the drawback of the SCF approach is that it can be computationally demanding to converge (see below).

The second approach is to treat the Drude particles dynamically from an extended Lagrangian perspective. In practice, a small mass, m_D , is ascribed to the Drude particles (typically 0.4 amu) while the mass of the parent atom, m_i , is decreased by a corresponding amount to preserve the total mass of the Drude-atom pair. To approximate the dynamics of the system on the SCF energy surface, Lamoureux and Roux introduced a dual-thermostat extended Lagrangian algorithm.^[31] to perform the numerical integration of the equation of motion. In this integration scheme based on a velocity-Verlet algorithm, the equations of motion use the absolute atomic and Drude coordinates ($\mathbf{r}_i$ and $\mathbf{r}_{D,i}$ respectively) to act on the Drude-atom center of mass ($\mathbf{R}_i$) and the Drude-atom displacement ($\mathbf{d}_i$) such that the forces are calculated as:

$$\mathbf{F}_{\mathbf{R},i} = -\frac{\partial U}{\partial \mathbf{r}_i} - \frac{\partial U}{\partial \mathbf{r}_{D,i}} \quad (3)$$

$$\mathbf{F}_{\mathbf{d},i} = -\left(1 - \frac{m_D}{m_i}\right) \frac{\partial U}{\partial \mathbf{r}_{D,i}} + \left(\frac{m_D}{m_i}\right) \frac{\partial U}{\partial \mathbf{r}_i} \quad (4)$$

To approximate the SCF surface, the Drude oscillators are coupled to a low-temperature Nosé- Hoover thermostat,^[36,37] separate from that of the real atoms, that defines a relative temperature between the Drude particles and their parent atoms. The temperature of the Drude thermostat (T_*) is set well below that of the real atoms (T , for the “physical” thermostat), such that $T_* \ll T$. Typically, for a system simulated at ambient temperature and m_D of 0.4 amu, T_* is set to 1 K. Thus, the equations of motion in the canonical ensemble are:

$$m_i \ddot{\mathbf{R}}_i = \mathbf{F}_{\mathbf{R},i} - m_i \dot{\mathbf{R}}_i \dot{\eta} \quad (5)$$

$$m_i' \ddot{\mathbf{d}}_i = \mathbf{F}_{\mathbf{d},i} - m_i' \dot{\mathbf{d}}_i \dot{\eta}_* \quad (6)$$

$$Q \ddot{\eta} = \sum_j m_j \dot{R}_j^2 - N_f k_B T \quad (7)$$

$$Q_*\ddot{\eta}_* = \sum_j m'_j \ddot{d}_j^2 - N_{f*} k_B T_* \quad (8)$$

where the forces on $\mathbf{R}_i$ and $\mathbf{d}_i$ are given by Eq. 3 and 4 and m'_i is the reduced mass of the Drude-atom pair, $m'_i = (1 - m_D/m_i)$. Atom indices i and j run over all atoms in the system. N_f and N_{f*} are the number of degrees of freedom in the thermostats coupled to atoms and Drude particles, respectively. The friction coefficients η and η_* (calculated from the forces acting on the thermostats, Eq. 7 and 8) are used to scale center-of-mass and displacement velocities of the Drude-atom pairs. Integration in center-of-mass and displacement coordinates is the same as integration in normal Cartesian coordinates of the atoms, so the velocity Verlet integrator is unchanged in this respect. The integration of the thermostat variables is done by transforming absolute coordinates $\mathbf{r}_i$ and $\mathbf{r}_{D,i}$ into $\mathbf{R}_i$ and $\mathbf{d}_i$, then transforming back when updating the velocities of the particles in the system. The integrator thus consists of propagation of the velocity Verlet term and the multi-step Nosé-Hoover terms, according to Martyna et al.^[38] This integration scheme was implemented in GROMACS version 4.5^[34] and the present code is an extension of it.

IMPLEMENTATION DETAILS

Additions to Topology Generation

The GROMACS program `pdb2gmx` processes a coordinate file, adds missing hydrogenatoms, and writes a topology that is used for subsequent MD simulations. The input files for this process are a coordinate file from the user and several library files in the force field that are packaged with GROMACS, most importantly the residue topology database (file extension `.rtp`) and termini database (file extension `.tdb`). These files define the residues available in the force field and their atomic connectivity (`.rtp`) and the modifications that can be made to them in the case of terminal residues (`.tdb`). To implement the Drude-2013 force field, these file formats have been extended to process Thole screening of neighboring dipoles,^[39] anisotropic polarizability,^[23] and virtual sites. While GROMACS has supported Thole screening and virtual site construction during MD integration for many years, these interaction types were not supported in `.rtp` and `.tdb` files for input processing and topology construction until now. Anisotropic polarizability in GROMACS was previously specific to water,^[14] but a new, generalized function has been written to handle any functional group characterized by anisotropic polarization. In addition, `pdb2gmx` has been extended to support construction of Drude particles and any missing lone pairs (virtual sites) specified in the `.rtp` entry for a given residue. The force field as currently implemented in GROMACS supports only protein, phosphatidylcholine-containing lipids, DNA, a subset of carbohydrates and monovalent ions at this time, but will be extended in the future to include multivalent ions as well as additional biomolecules such as RNA, lipids, and carbohydrates. The user can also provide a macromolecular structure that has already been processed in the Drude Prepper within CHARMM-GUI^[40] as input for `pdb2gmx`.

New Thermostat Functions

New algorithms have been implemented within the framework of the existing Nosé-Hoover thermostat^[36,37] in GROMACS. The differences are transparent to the user, but rather than applying the normal velocity scaling of the Nosé-Hoover thermostat to Drude-atom pairs, the velocity scaling is done as described above, in terms of center-of-mass and relative velocities. Non-polarizable (hydrogen) atoms have their velocities scaled in the conventional manner. The thermostat variables are updated in accordance with the algorithm of Martyna et al.^[38]

The Drude “Hard Wall” Constraint

A practical consideration in the simulation of polarizable systems is the integration time-step that can be used. In the past, polarizable systems were limited to relatively short sub-fem to second time-steps, which negatively impacted the efficiency of simulating such systems when similar systems with additive force fields (and covalent bond constraints) can be run with a time step of 2 fs. The main problem in polarizable simulations based on the Drude force field is the so-called “polarization catastrophe”^[41] in which the Drude particle is displaced significantly away from its parent atom. While the atom-Drude distance is typically on the order of ~ 0.1 Å, rare excursions to larger distances introduce large and sudden forces that lead to simulation instabilities and failure. To address this issue, Chowdhary et al. introduced a reflective “hard wall” to limit the maximum displacement of the Drude particle from its parent atom.^[22] A user-defined distance limit is set (typically 0.2 – 0.25 Å), and if the Drude particle reaches this limit, the hard wall constraint acts to reflect the Drude particle along the atom-Drude bond vector back towards the parent atom. The positions and velocities of the Drude particle and its parent atom are scaled to account for this update, as described previously.^[22] Use of this hard wall constraint allows for a stable integration time step of 1 fs without affecting the statistical properties of the simulated systems. The hard wall constraint has been implemented in GROMACS to allow efficient simulations using the Drude-2013 force field.

RESULTS AND DISCUSSION

System Descriptions

To evaluate the stability and performance of the new code, we performed simulations of two systems: a box of 1728 SWM4-NDP water^[16] molecules (8640 particles, including real atoms, Drude oscillators, and lone pairs) and ubiquitin (PDB 1UBQ,^[42] with all residues protonated according to their dominant form at pH 7, which yields a net neutral protein) in a box of SWM4-NDP water (6353 water molecules, a total of 33,837 particles in the system). Validation of the current code requires a simple system with easily definable properties, and the SWM4-NDP water box provides a convenient model, as well as a test of the scalability of the code, given its small size. The ubiquitin system serves as a typical case for running biomolecular MD simulations. The SWM4-NDP water box was prepared in CHARMM by equilibrating it under an NPT ensemble (298 K and 1.0 atm) for 100 ps using extended Lagrangian dynamics to achieve a suitable density. This system was subsequently simulated in GROMACS for 1 ns under an NVT ensemble at 298 K to compare its physical properties with those previously published using CHARMM.^[16] The configuration of the

ubiquitin system was taken from a previous study.^[25] CPU benchmarking was conducted using a 64-CPU AMD Opteron 6276 node with 2300-MHz processors and 2 GB of RAM per processor. Performance on a GPU was assessed with two Intel Xeon X5675 processors (12 total cores, 3.06 GHz) with an NVIDIA Tesla K20c GPU. Parallelization was achieved using OpenMP shared-memory parallelization or the built-in GROMACS thread-MPI library.

Simulation run settings were kept as consistent as possible across the different simulation software, though available features and implementation details inherently vary. For CHARMM and NAMD simulations, van der Waals interactions were switched to zero from 10.0 – 12.0 Å, with neighbor lists updated within 16.0 Å. In GROMACS, the same switching was applied, but the neighbor list radius was tuned at runtime under the Verlet cutoff scheme,^[43] with a minimum value of 12.0 Å and a maximum allowable per-particle energy drift of 0.005 kJ mol⁻¹ ps⁻¹ (the default setting). Decreasing this value to 0.0001 kJ mol⁻¹ ps⁻¹ had no impact on the outcome of the simulations, so for performance reasons it was left at the default setting. All simulations used the particle mesh Ewald method^[44,45] for electrostatics and periodic boundary conditions were applied in all three spatial dimensions. Bonds involving hydrogen atoms were constrained and the time step for all simulations was 1 fs, though we note that a time step of up to 1.25 fs also leads to stable integration. Water molecules were kept rigid with SETTLE^[46] (in GROMACS and NAMD) or SHAKE^[47] (in CHARMM). All simulations included isotropic long-range correction to account for truncation of the van der Waals terms. For simulations in GROMACS with the SCF approach for updating Drude oscillator positions, the force tolerance for convergence was set to 1.0 kJ mol⁻¹ nm⁻¹ and the maximum number of SCF iterations per MD time step was set to 50.

Performance

Results of CPU benchmarking for the water and ubiquitin systems are shown in Figure 1. The performance of GROMACS surpasses the existing implementation of extended Lagrangian dynamics in both CHARMM and NAMD, as well as the previously available SCF method in GROMACS. The code exhibits near-linear scaling even to ~600 atoms/CPU in the ubiquitin system. Parallel efficiency in CHARMM using the recently developed DOMDEC feature^[48] degrades below ~1000 atoms/CPU (only improving from 1.09 ns/day on 32 CPU to 1.12 ns/day on 64 CPU for ubiquitin), and, while performance in NAMD continues to improve up to 64 CPU, it remains well below that of GROMACS and deviates significantly from linearity. At the highest degree of parallelization tested for SWM4-NDP (32 CPU, 360 atoms/CPU), GROMACS is 3.2 times faster than CHARMM and 2.1 times faster than NAMD. For the ubiquitin system on 64 CPU, GROMACS is 4.9 times faster than CHARMM and 1.6 times faster than NAMD. The extended Lagrangian algorithm is also approximately 4 times faster than the previously existing SCF approach, due to the fact that the SCF approach spends considerable time in the energy minimization of the Drude oscillators, requiring multiple force evaluations per MD time step. For the SWM4-NDP systems, the SCF approach required an average of 5 iterations per step to converge the forces on the Drude oscillators below 1.0 kJ mol⁻¹ nm⁻¹ and 7 iterations to converge below 0.1 kJ mol⁻¹ nm⁻¹. The stricter convergence criterion led to no discernible improvement in

the accuracy of the simulations (see below); for this reason, $1.0 \text{ kJ mol}^{-1} \text{ nm}^{-1}$ is regarded as an adequate tolerance for SCF convergence. For the ubiquitin system, an average of 7 iterations was required to converge below $1.0 \text{ kJ mol}^{-1} \text{ nm}^{-1}$. In all cases, the SCF approach converged the forces below the tolerance at every MD step.

It should be noted that the parallelization scheme used by GROMACS depends on the number of processors utilized during the run. On fewer than 8 processors, domain decomposition^[33] is not invoked, and parallelization is achieved using OpenMP. The resulting performance should be interpreted in light of this fact, as the nonlinear increase in performance in moving from 4 to 8 CPU indicates that there is a considerable baseline enhancement in throughput due to domain decomposition over OpenMP. Further, when running on more than 16 CPUs, GROMACS allocated 8 CPU specifically for calculating PME mesh forces. The remaining CPUs were dedicated to particle-particle (PP) forces. This PP/PME balance is a significant factor in the excellent performance of GROMACS.

Another important consideration is the performance of the Drude extended Lagrangian simulations relative to similarly sized systems simulated under additive force fields. The results of the SWM4-NDP water simulation in GROMACS were compared against systems of 2160 TIP4P^[49] and 2880 TIP3P^[49] water molecules (8640 particles each, equivalent to that of the SWM4-NDP system). SWM4-NDP, TIP3P, and TIP4P are similar in that all three are rigid (i.e. have constrained internal geometries) and TIP4P has a virtual site whose position is constructed at every time step based on the coordinates of the real atoms. SWM4-NDP has an additional harmonic bond between the oxygen atom and its Drude particle that makes the force calculation inherently somewhat more expensive, but the comparison is reasonable in terms of the number of particles. Simulations of ubiquitin in water were also conducted using the CHARMM36 additive force field^[50] in TIP3P water, as an additional frame of reference. Though the number of particles in this system (20290) is inherently smaller, it is a useful comparison for scalability. As with the polarizable systems, the time step was set to 1 fs. Performance is given in terms of MD steps per second in Table 1. The outcome indicates that the polarizable simulations are approximately 3 times slower on CPU than an additive system with the same number of physical atoms. The relative speed-up with increasing processor count in GROMACS is equivalent between additive and polarizable systems, roughly linear to 16 CPU, and between 1.5 – 1.8× speedup with further doubling of the core count, indicating that the extended Lagrangian code does not degrade parallel efficiency.

For simulations run on a GPU, the extended Lagrangian does not show a significant enhancement in performance as in the case of the additive systems, indicating that further optimizations will be required to better harness the computing power of the GPU for computing nonbonded forces and balance the CPU and GPU workloads. For both SWM4-NDP and ubiquitin systems, the performance of the extended Lagrangian and SCF algorithms is approximately equivalent on the GPU.

Validation

The integrity of the GROMACS implementation of the extended Lagrangian dynamics was assessed by comparing properties of the SWM4-NDP water system with those produced by

CHARMM over the course of 10-ns simulations. The results of these simulations are summarized in Table 2 and indicate that all approaches generate consistent outcomes.

The dual Nosé-Hoover thermostat implementation was validated by examining the temperature and kinetic energy CHARMM differing by only 0.03% for both quantities. The temperature distributions for the “physical” thermostats overlap almost exactly for CHARMM and GROMACS (Figure 2), independent of the number of subdivided time steps used for updating thermostat variables (see Supporting Information). Thus, the two programs produce essentially identical results in terms of velocities. Similarly, by performing single-point energy evaluations on identical configurations, the energy terms in CHARMM and GROMACS differed by no more than 0.2%, indicating strong agreement between GROMACS and CHARMM with respect to force and energy evaluation.

CONCLUSIONS

In the present work, we have described the implementation of a Drude polarizable force field in the GROMACS simulation package. The GROMACS topology format has been extended to include anisotropic polarization required by this force field, and topology generation code has been updated to include Thole screening, anisotropic polarization, and virtual sites in force field library files. Most notably, the extended Lagrangian algorithm for integrating Drude positions has been implemented in the velocity Verlet integrator for efficient simulations of polarizable systems in the canonical ensemble. The code can be run on either CPU or GPU hardware and in parallel using either OpenMP or GROMACS domain decomposition. Future work will be required to make the extended Lagrangian algorithms compatible with the isothermal-isobaric (NPT) ensemble commonly used in biomolecular simulations. Further, the through-space Thole screening that CHARMM and NAMD use to screen interactions between water and divalent metal ions^[41] is not currently available. As such, only monovalent ions are supported in the present GROMACS implementation of the Drude-2013 force field, but work is ongoing to implement this feature for a future release.

The GROMACS extended Lagrangian code outperforms both the CHARMM and NAMD implementations in the two model systems studied here, across the entire range of processors. At the highest level of parallelization, at fewer than ~600 atoms/processor, GROMACS simulations are between 60–110% faster than NAMD, with the greatest gains evident in smaller systems. The code reproduces the energies from CHARMM and therefore represents an equivalent implementation with the added benefit of considerably improved (3–5 times faster) performance. The GROMACS extended Lagrangian code is also approximately 4 times faster than the previously available SCF algorithm, due to the time spent performing energy minimization on the Drude oscillators in the latter approach.

The code described here has been uploaded to the master branch of the GROMACS developmental code repository and will be included in a future release of the GROMACS package, or it can be obtained from <https://gerrit.gromacs.org> following instructions found at http://www.gromacs.org/Developer_Zone/Git/Gerrit.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

ACKNOWLEDGMENTS

The authors thank the GROMACS development team, especially Berk Hess, Mark Abraham, and Michael Shirts for helpful discussions during the development of the code, Paul van Maaren and Wojciech Kopec for testing and troubleshooting, and Jing Huang for helpful discussions about the CHARMM implementation of the extended Lagrangian algorithms and providing the coordinates of the ubiquitin system. This work has been supported by NIH grants GM072558 (B.R. and A.D.M.), GM070855 and GM051501 (A.D.M.), and F32GM109632 (J.A.L.). Computing resources were provided by the Computer-Aided Drug Design Center at the University of Maryland, Baltimore.

REFERENCES

1. Warshel A, Levitt M. J. Mol. Biol. 1976; 103:227–249. [PubMed: 985660]
2. Liu Y-P, Kim K, Berne BJ, Friesner RA, Rick SW. J. Chem. Phys. 1998; 108:4739–4755.
3. Kaminski GA, Stern HA, Berne BJ, Friesner RA, Cao YX, Murphy RB, Zhou R, Halgren TA. J. Comput. Chem. 2002; 23:1515–1531. [PubMed: 12395421]
4. Ren P, Ponder JW. J. Comput. Chem. 2002; 23:1497–1506. [PubMed: 12395419]
5. Xie W, Pu J, MacKerell AD Jr, Gao J. J. Chem. Theory Comput. 2007; 3:1878–1889. [PubMed: 18958290]
6. Jorgensen WL, Jensen KP, Alexandrova AN. J. Chem. Theory Comput. 2007; 3:1987–1992. [PubMed: 21132092]
7. Shi Y, Xia Z, Zhang J, Best R, Wu C, Ponder JW, Ren P. J. Chem. Theory Comput. 2013; 9:4046–4063. [PubMed: 24163642]
8. Rick SW, Stuart SJ, Berne BJ. J. Chem. Phys. 1994; 101:6141–6156.
9. Rick SW, Berne BJ. J. Am. Chem. Soc. 1996; 118:672–679.
10. Stern HA, Rittner F, Berne BJ, Friesner RA. J. Chem. Phys. 2001; 115:2237–2251.
11. Patel S, Brooks CL. J. Comput. Chem. 2004; 25:1–16. [PubMed: 14634989]
12. Patel S, MacKerell AD Jr, Brooks CL. J. Comput. Chem. 2004; 25:1504–1514. [PubMed: 15224394]
13. Drude, P.; Millikan, RA.; Mann, RC. The Theory of Optics. New York: Longmans, Green, and Co.; 1902.
14. van Maaren PJ, van der Spoel D. J. Phys. Chem. B. 2001; 105:2618–2626.
15. Kunz A-PE, van Gunsteren WF. J. Phys. Chem. A. 2009; 113:11570–11579. [PubMed: 19663490]
16. Lamoureux G, Harder E, Vorobyov IV, Roux B, MacKerell AD Jr. Chem. Phys. Lett. 2006; 418:245–249.
17. Yu W, Lopes PEM, Roux B, MacKerell AD Jr. Chem. Phys. Lett. 2013; 418:245–249.
18. 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]
19. Savelyev A, MacKerell AD Jr. J. Comput. Chem. 2014; 35:1219–1239. [PubMed: 24752978]
20. Savelyev A, MacKerell AD Jr. J. Phys. Chem. B. 2014; 118:6742–6757. [PubMed: 24874104]
21. Patel DS, He X, MacKerell AD Jr. J. Phys. Chem. B. 2015; 119:637–652. [PubMed: 24564643]
22. Chowdhary J, Harder E, Lopes PEM, Huang L, MacKerell AD Jr, Roux B. J. Phys. Chem. B. 2013; 117:9142–9160. [PubMed: 23841725]
23. Harder E, Anisimov VM, Vorobyov IV, Lopes PEM, Noskov SY, MacKerell AD Jr, Roux B. J. Chem. Theory Comput. 2006; 2:1587–1597.
24. Huang J, MacKerell AD Jr. Biophys. J. 2014; 107:991–997. [PubMed: 25140435]
25. Huang J, Lopes PEM, Roux B, MacKerell AD Jr. J. Phys. Chem. Lett. 2014; 5:3144–3150. [PubMed: 25247054]

26. Lemkul JA, Savelyev A, MacKerell AD Jr. *J. Phys. Chem. Lett.* 2014; 5:2077–2083. [PubMed: 24976900]
27. Savelyev A, MacKerell AD Jr. *J. Phys. Chem. Lett.* 2014; 6:212–216. [PubMed: 25580188]
28. 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, Hodoscek 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]
29. 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]
30. Jiang W, Hardy DJ, Phillips JC, MacKerell AD Jr, Schulten K, Roux B. *J. Phys. Chem. Lett.* 2011; 2:87–92. [PubMed: 21572567]
31. Lamoureux G, Roux B. *J. Chem. Phys.* 2003; 119:3025–3039.
32. van der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJC. *J. Comput. Chem.* 2005; 26:1701–1718. [PubMed: 16211538]
33. Hess B, Kutzner C, van der Spoel D, Lindahl E. *J. Chem. Theory Comput.* 2008; 4:435–447.
34. Pronk S, Páll S, Schulz R, Larsson P, Bjelkmar P, Apostolov R, Shirts MR, Smith JC, Kasson PM, van der Spoel D, Hess B, Lindahl E. *Bioinformatics.* 2013; 29:845–854. [PubMed: 23407358]
35. Swope WC. *J. Chem. Phys.* 1982; 76:637–649.
36. Nosé S. *J. Chem. Phys.* 1984; 81:511–519.
37. Hoover WG. *Phys. Rev. A: At. Mol. Opt. Phys.* 1985; 31:1695–1697.
38. Martyna GJ, Tuckerman ME, Tobias DJ, Klein ML. *Mol. Phys.* 1996; 87:1117–1157.
39. Thole BT. *Chem. Phys.* 1981; 59:341–350.
40. Jo S, Kim T, Iyer VG, Im W. *J. Comput. Chem.* 2008; 29:1859–1865. [PubMed: 18351591]
41. 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]
42. Vijay-Kumar S, Bugg CE, Cook WJ. *J. Mol. Biol.* 1987; 194:531–544. [PubMed: 3041007]
43. Verlet L. *Phys. Rev.* 1967; 159:98–103.
44. Darden T, York D, Pedersen L. *J. Chem. Phys.* 1993; 98:10089–10092.
45. Essmann U, Perera L, Berkowitz ML, Darden T, Lee H, Pedersen LG. *J. Chem. Phys.* 1995; 103:8577–8593.
46. Miyamoto S, Kollman PA. *J. Comput. Chem.* 1992; 13:952–962.
47. Ryckaert JP, Ciccotti G, Berendsen HJC. *J. Comp. Phys.* 1977; 23:327–341.
48. Hynninen A-P, Crowley MF. *J. Comput. Chem.* 2014; 35:406–413. [PubMed: 24302199]
49. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML. *J. Chem. Phys.* 1983; 79:926–935.
50. Best RB, Zhu X, Shim J, Lopes PEM, Mittal J, Feig M, MacKerell AD Jr. *J. Chem. Theory Comput.* 2012; 8:3257–3273. [PubMed: 23341755]
51. Yeh I-C, Hummer G. *J. Phys. Chem. B.* 2004; 108:15873–15879.

Figure 1.

CPU benchmarking results for CHARM, NAM, and the new GROMACS implementation of the extended Lagrangian algorithm for the two test systems. The performance of the existing SCF code in GROMACS is also included for comparison. Performance is given in ns/day based on a 1-fs time step.

Figure 2. Temperature distributions for the “physical” thermostat in simulations with CHARMM and GROMACS. The value of “nsteps” is the number of subdivided integration time steps for updating thermostat variables. The reference temperature (298.15 K) is indicated by the dashed vertical line.

Table 1

Simulation performance (steps/s) for polarizable and additive systems on CPU (64-core AMD Opteron 6276, 2300 MHz with 2 GB of RAM per core) and GPU (NVIDIA Tesla K20c with two 6-core Intel Xeon X5675 processors, 3.06 GHz and 80 GB total RAM)

System	Software	Number of Processors						12+GPU
		2	4	8	16	32	64	
SWM4-NDP	CHARMM	3.59	6.59	10.58	30.99	42.96		
	NAMD	9.14	17.68	31.63	51.10	66.74		
	GROMACS	13.95	22.33	49.37	91.72	138.62		79.37
	GROMACS (SCF)	3.49	6.81	12.03	22.81	36.65		86.76
TIP3P	GROMACS	42.16	78.77	152.65	260.17	404.12		792.64
TIP4P	GROMACS	34.17	65.87	131.41	231.86	345.87		831.95
Ubiquitin (Drude)	CHARMM	0.66	1.67	3.91	7.53	12.60	13.00	
	NAMD	1.68	4.72	8.24	17.63	29.67	40.21	
	GROMACS	3.31	5.32	11.73	22.46	35.00	63.86	18.76
	GROMACS (SCF)	0.64	1.26	2.32	4.53	6.84	14.74	17.64
Ubiquitin (CHARMM36)	GROMACS	14.73	28.87	60.79	109.18	170.92	245.18	384.67

Table 2

Bulk properties (molecular volume, V_m ; diffusion coefficient, D ; dipole moment, μ ; change in internal energy upon liquefaction, u ; and density, ρ) from simulations of 1024 SWM4-NDP water molecules in GROMACS compared to reference values from Lamoureux et al.^[16] Diffusion coefficients have been corrected according to the method of Yeh and Hummer.^[51]

Property	Reference	Extended Lagrangian	SCF ^[a]	
			1.0 kJ mol ⁻¹ nm ⁻¹	0.1 kJ mol ⁻¹ nm ⁻¹
V_m (Å ³)	29.94	29.96 ^[b]	29.96 ± 0.001	29.94 ± 0.001
D (× 10 ⁻⁵ cm ² s ⁻¹)	2.75	2.66	2.53	2.74
μ	2.46	2.46 ± 0.17	2.46 ± 0.17	2.46 ± 0.17
u (kcal mol ⁻¹)	-9.923	-9.93 ± 0.05	-9.95 ± 0.05	-9.94 ± 0.05
ρ (g cm ⁻³)	0.997	0.998 ^[b]	0.998	0.999

^[a]Two simulations were performed with the values listed as the maximum allowable gradient for SCF convergence.

^[b]Simulations with extended Lagrangian dynamics were carried out under an NVT ensemble following NPT equilibration with the SCF approach; thus V_m and ρ are fixed for this simulation.