

Performance Tuning of Polarizable Gaussian Multipole Model in Molecular Dynamics Simulations

Zhen Huang, Yongxian Wu, Yong Duan,* and Ray Luo*

Cite This: *J. Chem. Theory Comput.* 2025, 21, 847–858

Read Online

ACCESS |



Metrics & More

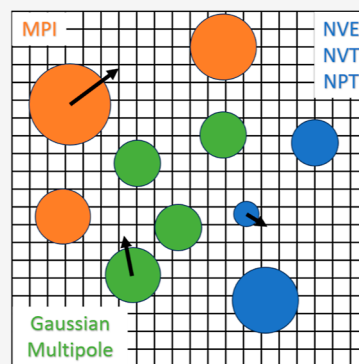


Article Recommendations



Supporting Information

ABSTRACT: Molecular dynamics (MD) simulations are essential for understanding molecular phenomena at the atomic level, with their accuracy largely dependent on both the employed force field and sampling. Polarizable force fields, which incorporate atomic polarization effects, represent a significant advancement in simulation technology. The polarizable Gaussian multipole (pGM) model has been noted for its accurate reproduction of ab initio electrostatic interactions. In this study, we document our effort to enhance the computational efficiency and scalability of the pGM simulations within the AMBER framework using MPI (message passing interface). Performance evaluations reveal that our MPI-based pGM model significantly reduces runtime and scales effectively while maintaining computational accuracy. Additionally, we investigated the stability and reliability of the MPI implementation under the NVE simulation ensemble. Optimal Ewald and induction parameters for the pGM model are also explored, and its statistical properties are assessed under various simulation ensembles. Our findings demonstrate that the MPI-implementation maintains enhanced stability and robustness during extended simulation times. We further evaluated the model performance under both NVT (constant number, volume, and temperature) and NPT (constant number, pressure, and temperature) ensembles and assessed the effects of varying timesteps and convergence tolerance on induced dipole calculations. The lessons learned from these exercises are expected to help the users to make informed decisions on simulation setup. The improved performance under these ensembles enables the study of larger molecular systems, thereby expanding the applicability of the pGM model in detailed MD simulations.



1. INTRODUCTION

Polarizable force fields have emerged as indispensable tools for computer-aided molecular modeling, playing pivotal roles in the analyses of both biomolecular and materials systems. Primarily due to the lack of the nonlinear induction term, the traditional additive force fields, while effective for general purposes, often fall short in accurately modeling electronic polarization effects.^{1–3} In contrast, polarizable force fields dynamically adjust to simulate electronic responses to molecular environments, capturing interactions between charges, induced dipoles, and higher order moments like quadrupoles. This dynamic modeling yields a representation of the electronic effect that more closely mirrors quantum mechanical calculations.

Several strategies have been developed to implement polarization within force fields. These include Drude oscillators,^{4,5} which simulate polarization through spring-connected dummy particles; fluctuating charge models,⁶ where the charges on atoms change depending on their environment; induced dipole methods^{7–11} that calculate the dipole moment induced by the local electric field; and continuum dielectric approaches^{12,13} that treat the solvent implicitly as a continuous polarizable medium. The AMOEBA force field represents a sophisticated implementation of the Thole-style¹⁴ induced dipole approach, combining permanent

atomic multipoles, up to atomic quadrupoles, with induced dipoles, and its efficient implementation in Tinker-HP has demonstrated strong performance on modern computing architectures.^{15,16} Another approach is the Gaussian electrostatic model (GEM), which uses density fitting and Hermite Gaussian auxiliary basis sets to represent atomic electron densities and extend to higher angular momenta (multipoles).^{17–19} Recent developments in PME methods for Gaussian integral evaluation have improved its efficiency, enabling parallel implementations that scale effectively up to hundreds of CPUs. The GEM model has been implemented in AMBERTOOLS, with the new GEM water model demonstrating scalability and accuracy against both QM level data and experimental measurements.^{20–22}

Building upon the advances in Gaussian-based methods, the polarizable Gaussian Multipole (pGM) model represents a significant innovation.^{11,23} The key advantage of pGM is its

Received: October 11, 2024

Revised: December 17, 2024

Accepted: December 20, 2024

Published: January 8, 2025



ability to integrate all electrostatic terms in a coherent manner, including all short-range and long-range terms, bonded and nonbonded terms. Given that the induction energy is a nonlinear term, such capability can be critical for accurate modeling of the induction terms that play pivotal roles in facilitating transferability across different electrostatic and solvation environments. It utilizes induced dipoles and incorporates Gaussian-shaped multipoles including charge and dipole to accurately model intra- and intermolecular electrostatic interactions. The use of Gaussian distributions for multipoles provides a smooth and continuous representation of the electrostatic potential, which is particularly advantageous for simulating complex molecular interactions. Moreover, Gaussian functions inherently account for charge penetration effects, eliminating the need for additional corrective terms.²⁴ The coherent representation of all monopoles and multipoles using Gaussian-shaped functions allows “screening” of all short-range terms, including charge–charge, charge–multipole and multipole–multipole terms, and effectively circumvents the “polarization catastrophe” that can occur with point-induced dipole models, where the polarization response can become unphysically large when atoms form close contacts.^{25,26} In addition, the coherent treatment of the intra- and intermolecular electrostatic interactions in pGM can facilitate seamless integration with quantum mechanical calculations.

The pGM model has been developed through a series of work in recent years.^{11,27–34} Including the initial formulation, improved formulation together with corresponding Particle Mesh Ewald (PME) algorithm, charge and dipole fitting methods with PyRESP and PCMRESP. In addition, the pGM model has been optimized in algorithm, making it a solid and fast model that captures correct energy functional and has demonstrated stability under NVE (constant number, volume, and energy) ensemble simulations.

Given the above developments and building upon previous successful parallel implementations of Gaussian-based force fields,²⁰ the need for more efficient parallel implementation of pGM becomes apparent as we utilize extensive simulations to optimize parameters and evaluate the force field’s performance under various conditions. While other Gaussian-based approaches have demonstrated successful parallelization, pGM’s distinct functional form necessitates a specialized implementation to handle its unique requirements for higher moments and self-consistent iteration calculations. Previous studies employing the pGM model in 1 ns simulations under the NVE ensemble have successfully demonstrated energy convergence, suggesting the model’s stability and reliability. Nonetheless, to comprehensively assess the force field’s effectiveness and robustness, longer simulations across different simulation ensembles are imperative. Developing an MPI (message passing interface) parallelized version of the pGM model is therefore essential to enable the execution of these extensive simulations, paving the way for more detailed and extensive exploration of biomolecular and material systems. In addition, our next step of development is to implement the pGM method to GPU platforms. Since AMBER pmemd program has been successfully optimized on CUDA and other GPU platforms, our plan is to start from these programs.

In a typical molecular dynamics (MD) engine, the use of MPI parallelization can be primarily categorized into two broad strategies: atom-based and spatial-based decomposition, each tailored for different types of interactions within the

simulation.³⁵ In the atom-based decomposition each processor is assigned a specific subset of atoms. This approach is particularly effective for calculating short-range and bonding interactions because it allows each processor to focus on a localized group of atoms. Although this method enables efficient computation of local interactions and is simpler to implement, it requires intensive all-to-all communication. This communication pattern, where each processor needs to share information with every other processor, can become a significant bottleneck, especially as the number of processors increases. The communication overhead of this method is largely influenced by the dynamic nature of molecular interactions, where atom neighborhoods can frequently change, necessitating constant updates across processors.

On the other hand, the second strategy, spatial decomposition, is especially beneficial for managing long-range interactions, which are crucial in simulations that employ the PME algorithm.^{36–38} The PME algorithm operates by calculating forces in both real and reciprocal spaces, the latter often using a mesh-based approach facilitated by fast Fourier transforms (FFT). Depending on the FFT implementation, the reciprocal space calculations can be divided into slabs or blocks, each managed by different processors. This spatial division not only helps in localizing the data processing to specific regions but also significantly reduces the communication overhead by limiting the data exchange of the 3D FFT/inverse FFT to neighboring processors only.^{15,39}

In our prior study, we developed the PME implementation for the pGM model,¹¹ which is distinguished by its needs for handling higher moments and self-consistent iteration to calculate induction effects. The induced dipole of each atom is determined through the interactions with the permanent moments and other induced dipoles, a process that leverages error functions and Gaussian functions to describe the energy interactions. This implementation comprises three main components: direct summation, reciprocal summation, and correction for self-interactions. The direct summation is handled within a defined cutoff distance and involves calculating the gradient of the error function, aligning this process closely with traditional force field approaches. The reciprocal summation introduces a deviation from standard PME methods, incorporating a phase vector to adjust the energy functional. This adjustment requires adding specific factors to accommodate the grid charge and dipole distribution during each iteration of the induction process.

The complexity of the pGM model, especially due to the requirement for recalculating PME in every induction iteration, increases the communication overhead in the MPI environment. This aspect, alongside the implications of such communication demands, is further elaborated in the methodology section below. With the parallelization of the code, our numerical method and Ewald settings may differ from those used in the serial settings. In this study, we will also explore how communication between threads affects these numerical algorithms and performance of the model and provide optimal configurations for simulations.

Following the implementation of the MPI version of the pGM PME code, it is imperative to thoroughly evaluate the stability and precision of the model across various simulation ensembles. This evaluation is crucial not only for verifying the integrity of the MPI implementation but also for ensuring that the pGM model accurately mirrors real-world physical behaviors under different controlled conditions. In this study,

we carefully analyze the performance of the pGM model under *NVT* (constant number, volume, and temperature) and *NPT* (constant number, pressure, and temperature) ensembles, closely monitoring temperature, potential energy, and density. For each ensemble, crucial metrics such as temperature equilibration and energy stability are scrutinized to assess the robustness of the pGM implementation.

Finally, this series of simulations includes tests specifically designed to assess the impact of convergence tolerance on induced dipole calculations and the effects of varying time step sizes. These tests are crucial for optimizing the model's computational efficiency without sacrificing the accuracy of the results. For example, by implementing a relaxed convergence criterion, we aim to significantly reduce the computational overhead while maintaining a high level of accuracy, especially in long-time scale simulations where dipole convergence might otherwise become a bottleneck. Additionally, our analysis on time step variations provides critical insights into the model's performance with an increased time step.

Through these focused tests, we aim to establish a comprehensive understanding of the computational reliability and efficiency of the pGM model, ensuring it is suited for both academic research and practical applications in fields like chemistry, biochemistry, and material sciences.

2. METHODS

2.1. MPI Parallelization of pGM Model. Integrating MPI parallelization into the molecular dynamics engine of SANDER within AMBERTOOLS suite of programs for the pGM model involved careful structuring of data communications and parallelization of nonbonded interactions. Figure 1 illustrates

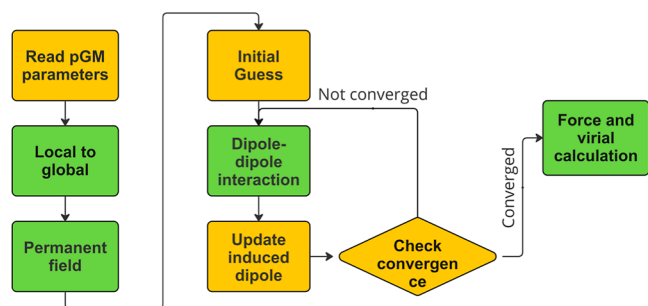


Figure 1. Workflowchart for the pGM model and the MPI communications applied within the pGM model. The yellow blocks indicate the operation completed within a single thread, while the green blocks represent operations performed locally in each thread. The induced dipole evaluation involves proposing an initial guess, followed by self-iteration, which includes calculating the dipole–dipole interactions and updating the induced dipole.

the workflow of the pGM model. The process begins with the master thread reading the corresponding parameters from the topology file and broadcasting them to all threads using `MPI_BCAST`. Because the permanent multipoles are expressed as covalent bond vectors, although the magnitude does not change in simulation, their directions do change. Thus, each thread, transforms the multipoles of the locally held atoms from the local framework to the global framework at every time step of integration following atomic coordinates updates and communicates these transformations using `MPI_GATHER`. Subsequently, the permanent charges and

dipoles are used to calculate the permanent field, which is communicated across threads.

Next, the master thread initiates the evaluation of the induced dipole, starting with the formation of an initial guess for the induced dipole. This is followed by calculating the dipole–dipole interaction, updating the search direction, proposing a new induced dipole, and checking for convergence. If convergence is achieved, the program proceeds to calculate the force and virial based on the total field. If not, it returns to the self-iteration loop to update the induced dipole until convergence is reached.

Different from our implementation in SANDER, PMEMD has a more mature system to optimize the code and parallelization. Essentially, we avoid using the group-to-group communications in the MPI library, but instead use the peer-to-peer communications. In this scheme, each processing element (PE) retains a set of lists of atomic information that need to be obtained from host PEs to perform the computations as well as a set of lists of partial calculated results that need to be sent to the host PEs. The information exchange takes place in an asynchronous manner via `MPI_Irecv()`, `MPI_Isend()`, and `MPI_Waitany()`.

It is worth noting that the field calculation and dipole–dipole interaction part, a matrix (dipole–dipole interaction tensor)–vector (field) multiplication, were segmented into three primary components: the direct, reciprocal, and self-correction parts. As shown in Figure 2, the direct part and self-

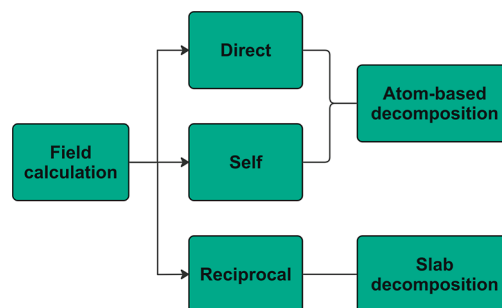


Figure 2. In the pGM model, the field calculation can be divided into three parts: the direct part, the reciprocal part and the self-correction part. The direct and self-correction parts use atom-based decomposition in MPI, while the reciprocal part is based on the spatial decomposition where in SANDER, we used slab decomposition.

correction part are based on atom decomposition, with some variations in strategy. For the direct part, since calculations are done for each pair of atoms, the decomposition is based on the cell list. Here, each thread tracks assigned cells and processes the atoms within them. In contrast, tasks like self-correction part and others using atom decomposition, such as local-to-global coordinate transformation, are managed by an atom list. Each thread maintains a local atom list to track which atoms it needs to process.

For the reciprocal part, the procedure involves first distributing the charges and dipoles onto a grid, followed by a backward FFT to convert the grid charge and grid dipole to the reciprocal space. For dipole–dipole interaction, the matrix–vector multiplication is simplified as a convolution operation, followed by a forward FFT to convert back to real space. The partitioning for the reciprocal part is based on spatial decomposition, where in SANDER the slab decomposition was implemented and in PMEMD the block decomposition

was enabled. Each thread is responsible for a slab or block, which is a natural fit for the grid mesh method. Notably, when building the slabs or blocks, we did not enable local communications, which is commonly used. Instead, we adopted a strategy where each slab or block slightly overlaps with its neighbors. This approach avoids the need for communication to obtain the information about charges and dipoles that can be extrapolated to the current slab/block but are located in neighboring slabs/blocks. This strategy trades some memory for time, and since only a small extension of the slab/block is required, it generally does not cause memory issues.

Communication complexities were further compounded by the different partitioning strategies used for direct and reciprocal interactions. Since the solver for the self-iteration is not distributed, it is necessary to communicate the field or the dipole–dipole interaction to the solver. As mentioned earlier, we avoid local communications between slabs by extending the slab, with each slab having an overlapping atom list. These overlapping atom lists in the reciprocal part made it challenging to efficiently package atom-based information. A streamlined approach involved consolidating this data into a global atom list and employing MPI_ALLREDUCE with MPI_SUM to manage the data more efficiently, thus reducing the complexity of data handling across all parallel threads. It is also worth noting that in PMEMD, we used an asynchronous peer-to-peer communication scheme that has the potential to notably reduce the communication overhead, especially when “local peek” is used for preconditioning in the induced dipole iterations that require only short-range (4–5 Å) direct sums.

2.2. Simulation Details. This study involved an extensive set of simulations using the pGM water model. All simulations were performed on a high-performance computing system equipped with four Intel Xeon Gold 6230 processors operating at a base frequency of 2.10 GHz. Each physical processor contains 20 cores with Intel Hyper-Threading Technology enabled, resulting in a total of 160 logical processors (40 per physical CPU). The system features 28160 KB (27.5 MB) of cache per CPU and supports advanced vector operations through AVX, AVX2, and AVX-512 instruction sets.

Our standard test system comprised 512 water molecules in a truncated-octahedral box with a dimension of 27 Å. To assess the scaling ability of the pGM model, we also included systems of varying sizes: 512, 1024, 2048, and 4096 water molecules. These systems were constructed using the PACKMOL package⁴⁰ and the tLEaP program from the AMBER 24 package.⁴¹

Simulations were performed using the AMBER 24 package, utilizing both the SANDER module from AMBERTOOLS²² and the PMEMD engine from AMBER 24.⁴¹ For MPI setups, SANDER and PMEMD were compiled with OPENMPI version 3.1.6,⁴² which was itself compiled using GCC version 11.3.0. For our standard 512 water system, the default setup employed 4 threads for MPI execution. To measure simulation efficiency, we conducted ten repetitions of 10 ps simulations on a dedicated quiet compute node, with all print outputs suppressed. The average time from these ten runs was recorded. To evaluate the performance of the pGM water model under various simulation ensembles, we collected 1 ns trajectories with outputs recorded every 1 ps.

The equations of motion were integrated using the leapfrog⁴³ algorithm with a time step of 1 fs. Water molecule geometry was constrained using the SHAKE⁴⁴ algorithm with a

tolerance of 10^{-6} . Electrostatic energy calculations were performed using the PME method for the pGM force field,¹¹ with a skin thickness of 2 Å for NVE and NVT simulations and 1 Å for NPT simulations. A cutoff of 9.0 Å was applied to limit the direct space sum, and van der Waals interactions were truncated at the same distance.

Analysis was performed using CPPTRAJ from the AMBER package, along with the Python SCIPY⁴⁵ and NUMPY⁴⁶ packages. Thermodynamic properties, including isothermal and isobaric heat capacity, and isothermal compressibility, were calculated following the definitions from our previous publication on the POL3 water model.⁴⁷ The self-diffusion coefficient was obtained from the velocity autocorrelation function.⁴⁸

3. RESULTS AND DISCUSSION

3.1. Scaling of pGM Model with Different Box Sizes of Water. First, we evaluated the scalability of the pGM model using various box sizes containing different numbers of pure water molecules. Specifically, we selected systems with 512, 1024, 2048, and 4096 water molecules in truncated octahedral periodic boxes, corresponding to box lengths of approximately 27, 35, 43, and 53 Å, respectively. This selection enabled a comprehensive analysis of the pGM model's scalability across a broad range of system sizes.

Figure 3 illustrates the scaling performance of SANDER. That of PMEMD is provided in the Supporting Information

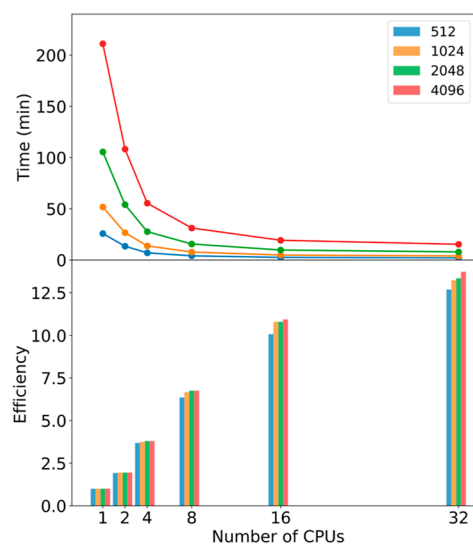


Figure 3. Scalability of the pGM MPI model for the water model across different system sizes with SANDER. In the figure, all simulations are conducted under the NVE ensemble, with each simulated for 10 ps. The efficiency is calculated as the time taken divided by the time taken for the serial code simulation.

(Figures S1 and S2). In this figure, the scaling efficiency is defined as the speed-up over the serial code. Figure S3 also shows the simulation speed, of both SANDER and PMEMD as measured by steps/second. As is typical, larger systems generally reach maximum efficiency with a higher number of computing threads, indicating that the optimal number of cores required for efficient scaling increases proportionally with the system and box size. This trend suggests that as the system size grows, leveraging more computational resources becomes necessary to maintain optimal performance and efficiency. For

comparison, detailed scaling performance data for both SANDER and PMEMD implementations is provided in the Supporting Information (Figure S3).

We would like to note that because polarizable force field simulations often require induced dipole iterations, we optimized the simulation programs such that the most time-consuming direct sum of the induced dipole calculations is performed using saved arrays, significantly reducing the computational demand. The direct sum part, which often has much better parallel scaling efficiency than the reciprocal part, no longer consumes the dominant portion of the simulation time. As a consequence, although such optimization notably improves the overall computational efficiency, the scaling data can have notably reduced parallel efficiency because relatively larger portion of the computer time is spent on comparatively less parallel parts.

Furthermore, Figure 4 presents a comparative analysis of the scaling performance between the pGM water model and the

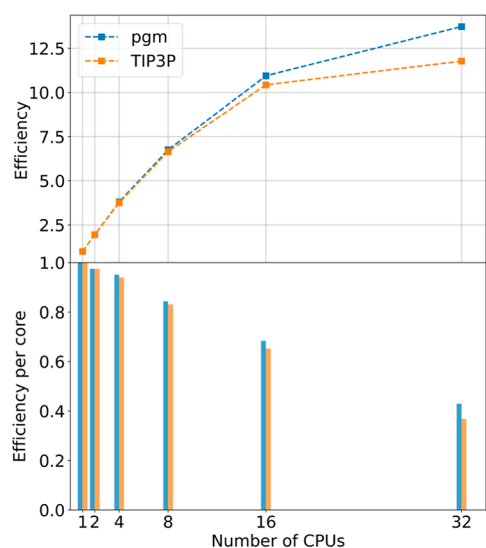


Figure 4. Efficiency of the pGM water model utilizing SANDER compared to the point-charge nonpolarizable TIP3P water model.

TIP3P water model using MPI. For this comparison, the 4096-water system was simulated. The figure compares the efficiency profiles of both the pGM and the nonpolarizable point-charge TIP3P water models. The results reveal that the pGM water model exhibits slightly better scaling behavior than the TIP3P water model. Additionally, since the pGM water model requires more time for simulation, its efficiency at different CPU counts is slightly higher than that of the TIP3P model. This is expected, as a more complex model may achieve better speedup. Given that significant proportion of the computation effort is spent on mutual induction calculations, these findings suggest that the pGM water model does not introduce a significant communication burden in the induced induction calculations compared to nonpolarizable models.

3.2. Stability of MPI Implementation of pGM. The second part of our study focuses on evaluating the stability of the MPI code in maintaining energy conservation during long NVE simulations. We compared the performance of the serial code with that of the MPI code by conducting NVE simulations and analyzing their energy conservation properties.

As illustrated in Figure 5, serial and MPI SANDER programs demonstrate similar energy conservation behavior, indicating

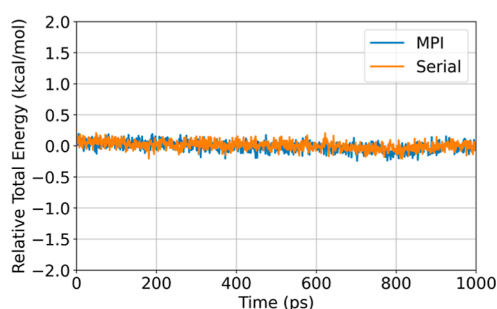


Figure 5. pGM simulation of 512 water model of 1 ns under NVE ensemble, the MPI and serial results are compared. Identical parameters were used in both simulations. Although there is small difference, both codes achieve energy conservation and thus proves the pGM MPI is stable and robust. The energy drift values for the are 1.13×10^{-4} kcal/mol ps mol in the MPI simulation and 9.33×10^{-5} kcal/mol ps in the serial simulation.

that the MPI implementation maintains the correct energy functional throughout the simulation. This is a crucial validation, ensuring that the parallelization does not adversely affect the fundamental physics of the simulation.

However, it is important to note that the MPI code may exhibit slight differences compared to the serial code due to the nature of parallel communication. For example, the MPI_ALLREDUCE operation can introduce a different order of operations and different number of operations as compared to serial program. In addition, as the data are computed locally in each thread, one needs to gather the data. The gathered data with addition/multiplication will introduce more operations compared to the serial code. In short simulations, the effect of the difference is minute, while in long simulations, the small difference can accumulate, potentially leading to a large divergence in trajectories exacerbated by the nonlinear potential energy function.

Our test shows that for a 1 ns simulation, these accumulated deviations remain negligible and do not compromise the conservation of energy. This demonstrates that the MPI code is robust and reliable for long time simulations for the pGM model, ensuring that energy conservation is preserved over significant time scales.

3.3. Ewald Parameters and MPI Version Multiorder Extrapolation Parameters. Throughout the development, we conducted extensive tests to evaluate the computational time required for different components of the pGM energy term evaluations. Specifically, we tracked the distribution of computational time across field calculations for a system with 512 water molecules, highlighting that the PME method, the primary component of the reciprocal calculation, consumes a significant portion of the computational time. Thus, minimizing the PME time can substantially reduce the overall simulation time.

To achieve optimal energy conservation and efficiently distribute dipoles onto the grid, we employed a high-order B-spline function in PME along with a large nfft value, generating a fine grid in reciprocal space. While this setup enhances accuracy, it also increases computational time. However, such high precision is not always necessary in practical simulations.

In this study, we systematically tested a range of B-spline function orders and various nfft values. These tests were conducted under NVE conditions to ensure energy conservation. The results confirmed that the selected B-spline

order and grid spacing are adequate for simulations. The outcomes, including energy drift values, are presented in Figures 6 and 7, while the simulation time data is provided in Tables 1 and 2.

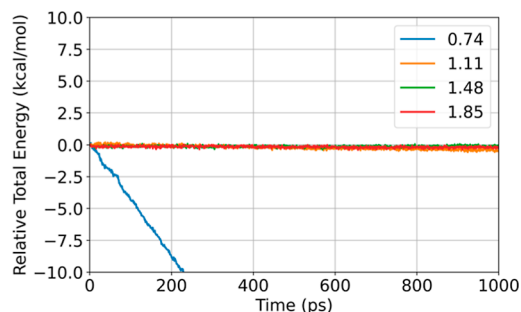


Figure 6. Energy drift with grid densities in pGM simulations of 512-water box of 1 ns under *NVE* ensemble. The label shows the density of the grid along one direction. The energy drift values are -4.65×10^{-3} , -5.05×10^{-4} , 8.00×10^{-6} , -9.00×10^{-6} kcal/mol ps for grid densities of 0.74, 1.11, 1.48, and 1.85 (per Å), respectively.

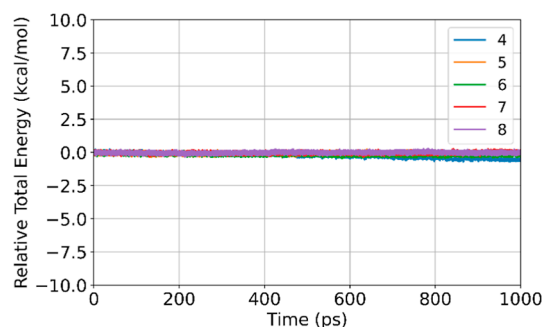


Figure 7. Energy drift with different B-spline orders in pGM simulations of 512 water model of 1 ns under *NVE* ensemble. The energy drift values are -6.03×10^{-3} , 3.39×10^{-4} , -9.61×10^{-5} , 6.93×10^{-4} , 4.23×10^{-5} kcal/mol ps for B-spline orders of 4, 5, 6, 7, and 8, respectively.

Table 1. Simulation Times for the 512-Water Box with Different Grid Sizes with 5th-Order B-Spline

grid size	time (s)
20×20×20	1465
30×30×30	1485
40×40×40	1574
50×50×50	1650

Table 2. Simulation Times for the 512-Water Box with Different B-Spline Orders and 30 × 30 × 30 Grid

B-spline order	time (s)
4	1395
5	1485
6	1605
7	1767
8	1948

As shown in Figures 6 and 7, we tested grid sizes ranging from $20 \times 20 \times 20$ to $50 \times 50 \times 50$ for the 512-water box ($27 \times 27 \times 27 \text{ Å}^3$) with a B-spline order of 6. We also tested B-spline function orders from 4 to 8 with a grid size of $30 \times 30 \times 30$. These figures indicate that with a grid size of $30 \times 30 \times 30$

and a B-spline order of 5, energy conservation is well satisfied, and the total energy drifted at a rate of -5.05×10^{-4} kcal/mol/ps. The grid density along one direction is 30/27, approximately 1.11 grid/Å. However, when B-spline was set of fourth order or grid density was lower than 1 grid/Å, the system total energy exhibited appreciable drift.

Tables 1 and 2 provide the CPU times for 10 ps simulations with these different settings, averaged over ten trials. Generally, lower grid density and lower B-spline order help reduce simulation time. As shown in Table 1, reducing grid size significantly decreases simulation time, particularly for FFT-related operations (backward FFT, convolution, forward FFT, scalar sum). Changes in B-spline order impact the time more significantly by reducing the time required for Fill B-spline coefficients, Fill charge/dipole grid, and scalar sum.

Notably, although energy drift at an order of 4 is not excessive, an order of 5 is required for accurate virial calculations in *NPT* simulations. Compared to our previous settings (B-spline order 8 with a $50 \times 50 \times 50$ grid), our new standard (order 5 with a $30 \times 30 \times 30$ grid) reduced the simulation time from 2064.33 to 1484.54 s, achieving approximately a 28% speedup while maintaining energy conservation.

In our previous work,³⁰ we introduced innovative methods for induction calculations, specifically the preconditioned conjugate gradient with local iterations (LIPCG) and multi-order extrapolation (MOE). These methods leverage previously computed induced dipoles to form new initial guesses, enhancing the efficiency of convergence. The LIPCG method excels at reducing the number of iterations and quickly finding the true solution when coupled with MOE, which effectively extrapolates historical information.

During the MPI development, LIPCG necessitates additional communications, as detailed in the Methods section. Consequently, different parameter settings for MOE may significantly impact computational speed. Here, we present our analysis of various combinations of MOE parameters using MPI in Table 3. We tested the *NVE* simulation using 4 and 16 CPU cores (for 512 water system and 4096 water systems respectively), and both cases showed results very similar to the serial code. In all scenarios, the first order with five previous steps, the second order with 2 to 4 previous steps, and the third order with 2 steps performed best. Among these, the

Table 3. Simulation Times (In Seconds) for the pGM Model with Various Multi-Order Extrapolation Settings^a

step	order = 1	order = 2	order = 3
512-water box with 4 threads			
2	2073	1797	1405
3	1890	1358	1520
4	1911	1362	1535
5	1799	1405	1558
6	1836	1450	1567
4096-water box with 16 threads			
2	5020	4892	4635
3	4407	4673	4925
4	4533	4766	3921
5	4253	4770	4146
6	4974	4961	4443

^aThe labels order = 1,2,3 represent the order of extrapolation, while the label Step indicates the number of steps used for the extrapolation.

third order with 2 history steps consistently yielded one of the best results across all test cases, consistent with our previous finding on the serial code.³⁰ This is expected as the communication time taken for MOE is minimal compared to the overall computation time even if the extra communication is needed.

3.4. Performance of pGM Model in Simulations Coupled to Thermal Baths. The pGM model has demonstrated robust performance for *NVE* simulations, as shown in our previous work³⁰ and earlier sections. However, in practical simulations, *NVE* ensemble is rarely used and *NPT* and *NVT* are often used instead. Thus, we would like to evaluate the performance in simulations in which the system is coupled to thermal baths. Here, we present our findings on the pGM model's performance for commonly used ensembles: *NVT* (constant number, volume, and temperature) and *NPT* (constant number, pressure, and temperature), using identical simulation conditions to those ensuring energy conservation in previous sections.

We conducted a series of simulations using the pGM model under *NVT* and *NPT* ensembles to evaluate its stability and accuracy in simulating thermodynamic properties. These simulations were designed to test the model's ability to correctly represent physical behaviors such as temperature and density fluctuations, as well as its capability to maintain equilibrium states. We selected the TIP3P water model for comparison to demonstrate that the pGM water model can produce property statistics similar to those of nonpolarizable models.

3.4.1. *NVT* Simulations. We tested the pGM model under the *NVT* ensemble with both Langevin⁴⁹ and Berendsen⁵⁰ thermostats. Figure 8 shows the temperature distribution for

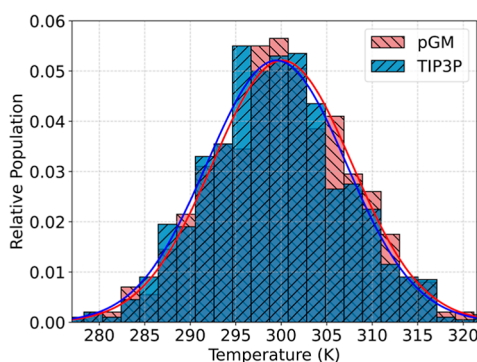


Figure 8. Histograms of temperature distributions of pGM and TIP3P models at 300 K with the Langevin thermostat under *NVT*. The mean and standard deviation are 300.11 and 7.66 for the pGM model, and 299.49 and 7.66 for the TIP3P model, respectively.

the Langevin thermostat, while the distribution for the Berendsen thermostat is available in Supporting Information Figure S4. As the Berendsen bath is a weak coupling scheme, its temperature distribution is narrower than that of the Langevin thermostat. Both distributions are comparable to TIP3P's distributions under the same simulation setups.

The potential energy distribution under the Langevin thermostat, which significantly contributes to thermodynamic properties, is detailed in Figure 9. The distributions for the Berendsen thermostat can be found in Figure S5. The pGM model's relative potential energy distribution has a slightly larger variance, leading to a different isothermal heat capacity.

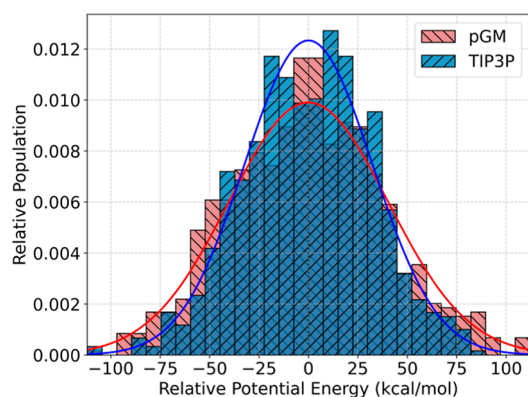


Figure 9. Histograms of relative potential energy distributions of pGM and TIP3P models at 300 K with the Langevin thermostat under *NVT*. The potential energy is relative to the mean value so that both distributions are centered at 0 kcal/mol. The standard deviations are 38.96 for the pGM water and 32.34 for the TIP3P water, respectively.

As shown in the *NPT* simulations below, the pGM model's larger heat capacity is in better agreement with experimental values.

Figure 10 compares the velocity correlation function (VCF) of the pGM model against that of the TIP3P model. The VCFs

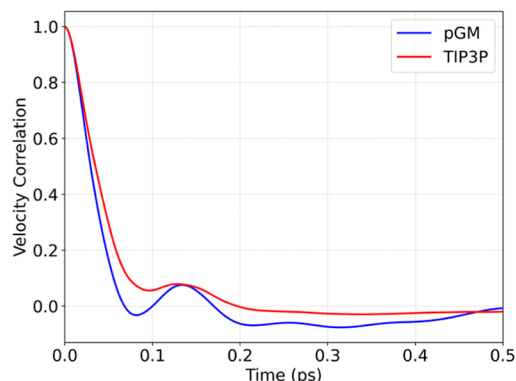


Figure 10. Comparison of the velocity correlation functions of pGM and TIP3P models.

of both models quickly decay to values around 0 with similar correlation times. The self-diffusion coefficients obtained from the VCFs are $2.42 \times 10^{-5} \text{ cm}^2/\text{s}$ and $5.39 \times 10^{-5} \text{ cm}^2/\text{s}$ for the pGM and TIP3P models, respectively. Compared to the experimental value $2.28 \times 10^{-5} \text{ cm}^2/\text{s}$, the pGM model value is very close and the TIP3P value is about twice as much due to the well-documented shallow energy well of the TIP3P model.

3.4.2. *NPT* Simulations. We tested the pGM model under the *NPT* ensemble using Berendsen and Langevin thermostats for temperature control and Berendsen and Monte Carlo barostats for pressure control. As the temperature and potential distributions are not significantly influenced by the barostat, they follow the same pattern observed under *NVT* conditions. Our primary focus here is on pressure-related properties, specifically the density distributions.

Figure 11 highlights the density distributions of pGM and TIP3P models under the Monte Carlo barostat. The distributions for the Berendsen barostat can be found in Figure S6. The standard deviations between the two water models are quite similar, indicating that the pGM model

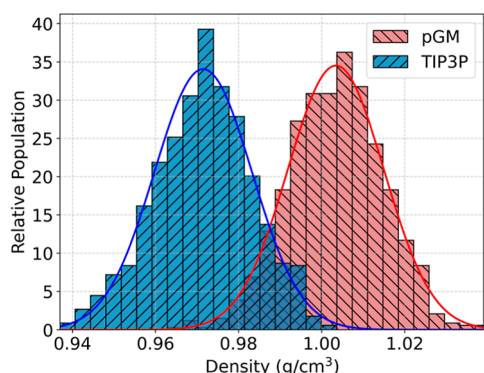


Figure 11. Comparison of the density distributions of pGM and TIP3P models at 300 K with Langevin thermostat and Monte Carlo barostat under *NPT*. Mean values are 1.006 g/cm³ and 0.977 g/cm³ for pGM and TIP3P waters, respectively. The variances are 0.012 for both water models.

behaves similarly to the TIP3P model in *NPT* simulations. The pGM model's density of 1.006 g/cm³ is closer to the experimental value of 0.997 g/cm³ at 300 K than that of TIP3P water (0.977 g/cm³). Interestingly, the pGM model's isothermal compressibility (related to volume fluctuation) of 50.61×10^{-6} /bar agrees better with the experimental value of 44.94×10^{-6} /bar than that of TIP3P water (58.50×10^{-6} /bar). Finally, the pGM model's energy fluctuation-related heat capacity of 18.20 cal/(mol K) also agrees well with the experimental value of 17.99 cal/(mol K), compared to 13.81 cal/(mol K) for the TIP3P model. More detailed comparisons with experimental data, along with temperature-dependent properties, will be reported in our other work on optimization of the rigid three-point pGM water model (manuscript in preparation).

3.5. Relaxed Convergence Tolerance and Larger Timestep in *NVT* and *NPT* Simulations. **3.5.1. Impact of Convergence Tolerance.** The accuracy and efficiency of molecular dynamics simulations involving polarizable force fields are significantly influenced by the convergence tolerance used in iterative induction calculations. In *NVE* simulations, strict convergence is essential to maintain energy conservation, as even minor deviations in induced dipole calculations can result in substantial drift in total energy. However, for *NVT* or *NPT* simulations, where the system is coupled to external temperature baths, exact energy conservation becomes less critical. Consequently, the convergence tolerance for induction calculations can be relaxed without introducing substantial artifacts into the simulation results.

Table 4 presents averages and standard deviations of key simulation properties, including potential energy, temperature, density, and volume, for various convergence tolerances in the

induction calculation. *NPT* simulations with the Langevin thermostat and the Monte Carlo barostat were conducted for illustrations. The results demonstrate that even with a loose convergence tolerance—one that would severely compromise energy conservation in *NVE* simulations—the average potential energy and temperature, along with their fluctuations, remain well within acceptable ranges when the temperature is regulated. This suggests that external coupling to temperature baths effectively mitigates the impact of less stringent convergence criteria. Furthermore, different tolerances yield highly similar average densities and volumes, as well as their fluctuations. The consistency across the tested convergence levels confirms that, while tight convergence is necessary for energy-conserving ensembles, relaxed tolerances (e.g., 10^{-3} and 10^{-2}) can be employed in simulations coupled with thermal baths without significantly affecting key thermodynamic properties.

The significance of these findings is self-evident. The dipole convergence process can be computationally expensive, particularly in large, polarizable systems. By allowing relaxed convergence criteria in simulations coupled with thermal baths, we can reduce computational overhead while maintaining accurate results. This approach represents a practical optimization, balancing computational efficiency with scientific accuracy, particularly in long-time scale simulations where induction calculations might otherwise become a performance bottleneck.

3.5.2. Impact of Timestep. The design of our rigid pGM water model presents an opportunity to explore the widely used longer integration time step of 2 fs, particularly in simulations coupled to thermostats. This strategy effectively doubles the temporal reach of simulations, offering a practical method to extend simulation duration without incurring excessive computational overhead.

However, the use of larger timesteps introduces significant challenges, especially for simulations utilizing polarizable force fields. These force fields are particularly sensitive to time step size due to their dependence on accurate calculations of induced dipoles. Larger timesteps may lead to simulation instability, as rapid changes occur in the molecular electrostatic environment. Consequently, longer timesteps are rarely used in *NVE* simulations.

Our previous research introduced an extrapolation scheme to predict initial induced dipoles based on information from prior timesteps, assuming that past configurations provide useful insights for initializing future states. However, increasing the time step may undermine this assumption. With a larger time step, molecular conformations undergo more significant changes between successive steps, potentially rendering previous configurations less relevant for predicting current

Table 4. Means and Standard Deviations of Potential Energy (PE), Temperature (Temp), Density, and Volume in *NPT* Simulations with Different Convergence Tolerances (τ) for the Induced Dipole Iteration^a

model	PE	temp	density	volume	D
$\tau = 10^{-2}$	−504668.32(40.49)	299.76(7.72)	1.008(0.012)	15197(174)	2.59(0.58)
$\tau = 10^{-3}$	−504666.13(40.02)	299.96(7.58)	1.008(0.012)	15203(176)	2.57(0.28)
$\tau = 10^{-4}$	−504666.44(40.14)	299.94(7.64)	1.006(0.012)	15232(179)	2.42(0.19)

^aThree production simulations of 1 ns each were collected with different random seeds with the Langevin thermostat and the Monte Carlo barostat. Additionally, the table includes the self-diffusion coefficient (*D*) averaged from five different short *NVT* simulations also in the Langevin thermostat. Potential energy is reported in kcal/mol, temperature in Kelvin, density in g/cm³, volume in Angstrom³, and self-diffusion coefficient in 10^{-5} cm²/s.

Table 5. Means and Standard Deviations of Potential Energy (PE), Temperature (Temp), Density, and Volume in *NPT* Simulations with Different Time Steps (Δt)^a

model	Δt (fs)	PE	temp	density	volume	<i>D</i>
pGM/ $\tau = 10^{-2}$	2	−504658.88(40.19)	299.48(7.65)	1.007(0.012)	15209(174)	2.29(0.29)
pGM/ $\tau = 10^{-3}$	2	−504655.09(39.88)	300.11(7.63)	1.007(0.011)	15215(172)	2.76(0.17)
TIP3P	1	−4858.26(34.40)	299.78(7.67)	0.977(0.012)	15681(195)	5.39(0.75)
TIP3P	2	−4848.01(34.69)	300.18(7.62)	0.977(0.012)	15688(200)	4.82(0.83)

^aSimulations were conducted for both pGM and TIP3P water models. τ is the induced dipole iteration tolerance. Each production simulation is 1 ns in length repeated three times with different random seeds with the Langevin thermal bath and Monte Carlo barostat. Additionally, the table includes the self-diffusion coefficient (*D*) averaged from five different short *NVT* simulations. The units of data reported are as follows: potential energy is in kcal/mol, temperature in Kelvin, density in g/cm³, volume in Angstrom³, and self-diffusion coefficient in 10^{−5} cm²/s.

states. This reduction in temporal resolution can lead to inaccuracies in initializing induced dipoles, further compounding the challenges introduced by the larger time step. Moreover, the use of the Langevin thermostat further complicates the initialization of future states due to the presence of random forces. Indeed, our tests indicate that utilizing information solely from the immediate previous step to initialize induced dipoles yields better results.

Given these considerations, a critical question arises regarding the feasibility of employing larger timesteps within pGM simulations. The potential implications on stability and accuracy must be carefully evaluated to determine if the benefits of increased computational efficiency outweigh the risks of diminished simulation fidelity. We have compiled simulation data in Table 5, presenting averages and standard deviations of key properties such as potential energy, temperature, density, and volume for *NPT* simulations. Simulations were conducted with relaxed induction tolerances of both 10^{−3} and 10^{−2} to evaluate simulation quality. Three different replicas of production simulations were collected for each testing condition to ensure robust statistics.

Comparison between Tables 5 and 4 reveals that the average properties in 2 fs simulations remain close to those of the corresponding 1 fs simulations, with standard deviations showing negligible differences between the two sets. As a reference, an identical comparison was conducted for TIP3P water, also shown in Table 5. Similar differences due to varying timesteps can be observed between the two water models. This suggests that the 2 fs time step can be effectively used with comparable simulation accuracy to that shown in TIP3P water simulations, at least for the tested Langevin thermostat.

Table 6 presents simulation times for 10 ps test runs with both 1 and 2 fs timesteps at different convergence tolerances.

Table 6. Times for 10 ps *NPT* Simulation with Langevin Thermostat and Monte Carlo Barostat^a

Δt (fs)	$\tau = 10^{-2}$	$\tau = 10^{-3}$	$\tau = 10^{-4}$
1	258	278	309
2	141	152	172

^aData were collected with SANDER.MPI with 8 CPU threads on a quiet compute node. τ is the induced dipole iteration tolerance. The simulation times are reported in seconds.

Regardless of convergence tolerance, the use of 2 fs reduces the simulation time to approximately 55% of that in the corresponding 1 fs simulation under otherwise identical conditions. Therefore, the 2 fs time step can be employed to significantly improve computational efficiency without substantial sacrifice in accuracy, similar to the widely used TIP3P

water model. This makes it suitable for long-time scale simulations where performance is critical.

4. CONCLUSION

This report documents our efforts to enhance the computational efficiency of pGM force field simulations using the MPI framework within the AMBER framework. Our comprehensive analysis demonstrates the model's performance under various simulation conditions, offering insights for molecular dynamics applications.

We evaluated the pGM model's scalability across different system sizes within the AMBER framework. Our analysis demonstrates reasonable parallel scaling for systems of increasing size, with larger systems benefiting from higher computational thread counts. The MPI-parallelized pGM model maintained accuracy and stability during extended simulations, while our systematic testing of Ewald and MOE parameters allowed for optimization of computational performance.

Our analysis extended to the pGM model's performance under *NVT* and *NPT* ensembles, demonstrating reliable temperature and pressure control with behavior comparable to the established TIP3P model. We also explored optimization strategies, including the implementation of a relaxed convergence criterion for induction calculations. This approach significantly reduced computational demands without compromising simulation accuracy, striking a crucial balance between efficiency and precision.

Furthermore, our investigation into time step variations yielded important insights. Notably, a 2 fs time step significantly improves computational efficiency while maintaining energy distribution stability and accuracy comparable to the 1 fs time step. This finding suggests that the 2 fs time step can be effectively utilized in pGM force field simulations, offering enhanced efficiency without compromising result reliability, particularly for long-time scale simulations.

In conclusion, our work demonstrates the pGM model's capability to handle molecular dynamics simulations with accuracy and stability within the AMBER framework. The model's adaptability to various simulation conditions, coupled with its optimized performance, provides a useful tool for researchers in the field. Future work will focus on GPU implementation, further optimizations, and exploring additional applications to expand its utility in computational chemistry and related disciplines.

APPENDIX

Simulation Protocols for pGM Simulations under *NVT* and *NPT*

To achieve optimal thermodynamic behavior while maintaining computational efficiency, the selection of an appropriate thermostat and barostat is crucial. Here, we outline the process for optimizing the pGM water model through extensive testing.

Thermostat Selection. The Berendsen thermostat, characterized by its weak coupling scheme, can lead to variances that deviate from true thermodynamic properties. In contrast, Langevin thermostat offers a more accurate variance in both temperature and potential energy distributions, though it is computationally slower. This is because initialization of dipoles during the induction calculation cannot fully utilize the noisy history information. The coupling parameter in the Berendsen thermostat provides an intuitive means of adjusting the coupling behavior. Detailed comparisons of these effects are presented in Table A1, which shows standard deviations in

Table A1. Effect of Thermostat Control Parameters on the Fluctuations of Temperature and Potential Energies^a

thermostat control parameter	σ_{temp}	σ_{PE}	time
Berendsen thermostat: τ_{autp} (ps)			
1.5	7.06	29.39	930
2.5	7.39	32.86	926
5.0	8.14	42.73	923
Langevin thermostat: γ_{in} (1/ps)			
1	7.66	38.33	1046
5	7.77	38.23	1059
10	7.70	38.71	1119

^aTemperature (temp) unit is in Kelvin and the potential energy (PE) unit is in kcal/mol. The simulation times were collected with SANDER.MPI with 4 CPU threads on a quiet compute node. The simulation times are reported in seconds.

both temperature and potential energy for different parameters. Table A1 shows that the Berendsen thermostat underestimates the fluctuations of temperature and potential energy while using a larger control parameter may help. Nevertheless, the simulation times are in general shorter than the simulations with the Langevin thermostat.

Barostat Selection. Our analysis of the pGM simulations suggests that the Berendsen weak coupling scheme may produce inaccurate thermodynamic properties. Alternatively, the Monte Carlo barostat avoiding the time-consuming calculations of the virial terms by switching between states, offers a potentially more accurate approach in simulating thermodynamic properties. The detailed analysis of the barostat's performance is provided in Table A2. Our analysis shows again that the fluctuation of density with the Berendsen barostat is smaller and may be improved by using a larger control parameter.

Based on our tests, we recommend the following protocol for simulations using pure water as an example. For thermodynamic properties under the *NVT* ensemble, start with the Langevin thermostat set to a collision frequency of 5.0 ps⁻¹ for heating over 200 ps, during which the system will gradually reach the target temperature. Following this, equilibrate with the Langevin thermostat at the same collision frequency for 400 ps. The restart file obtained at this stage is then used for production simulations. During production simulations, begin each run from the same restart file, applying

Table A2. Effect of Barostat Control Parameters on the Fluctuation of Density^a

Barostat control parameter	density	time
Berendsen barostat: τ_{autp}		
1.0	0.006	887
2.5	0.007	888
5.0	0.009	907
Monte Carlo barostat: mcbarint		
25	0.011	785
50	0.011	782
100	0.012	756

^aDensity is in the unit of g/cm³. The simulation times were collected with SANDER.MPI with 4 CPU threads on a quiet compute node. The simulation times are reported in seconds.

a different random seed for the initial velocities. Re-equilibrate using a Langevin thermostat with a collision frequency of 1.0 ps⁻¹ for 400 ps, followed by production simulations lasting 400–500 ps. Repeat this process at least 10 times to ensure statistical accuracy.

For thermodynamic properties under the *NPT* ensemble, start with the *NVT* restart file and use the Langevin thermostat with a collision frequency of 5.0 ps⁻¹ and the Berendsen barostat with a τ_{autp} of 1 ps for initial equilibration over 500 ps. This step ensures that the system pressure is quickly equilibrated. During production simulations, start each run from the same restart file, again using a different random seed for initial velocities. Re-equilibrate using the Langevin thermostat with a collision frequency of 1.0 ps⁻¹ and apply the Monte Carlo barostat with state changes attempted every 25 steps. Re-equilibrate for 500 ps, then perform production simulations for 400–500 ps, repeating the process 10 times.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jctc.4c01368>.

MPI scalability and performance of PMEMD (Figures S1 and S2) and comparisons between SANDER and PMEMD implementations of the pGM model (Figures S6), along with detailed analyses of temperature, potential energy, and density distributions compared with the TIP3P model under both Berendsen thermostat and barostat conditions (Figures S3–S5) (PDF)

AUTHOR INFORMATION

Corresponding Authors

Yong Duan – UC Davis Genome Center and Department of Biomedical Engineering, University of California, Davis, Davis, California 95616, United States; orcid.org/0000-0003-3793-5099; Email: duan@ucdavis.edu

Ray Luo – Chemical and Materials Physics Graduate Program, Departments of Molecular Biology and Biochemistry, Chemical and Biomolecular Engineering, Materials Science and Engineering, and Biomedical Engineering, University of California, Irvine, Irvine, California 92697, United States; orcid.org/0000-0002-6346-8271; Email: rluo@uci.edu

Authors

Zhen Huang – Chemical and Materials Physics Graduate Program, Departments of Molecular Biology and

Biochemistry, Chemical and Biomolecular Engineering, Materials Science and Engineering, and Biomedical Engineering, University of California, Irvine, Irvine, California 92697, United States; orcid.org/0009-0003-4478-8477

Yongxian Wu — Chemical and Materials Physics Graduate Program, Departments of Molecular Biology and Biochemistry, Chemical and Biomolecular Engineering, Materials Science and Engineering, and Biomedical Engineering, University of California, Irvine, Irvine, California 92697, United States

Complete contact information is available at:
<https://pubs.acs.org/10.1021/acs.jctc.4c01368>

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the research support from NIH (GM130367 to R.L.).

REFERENCES

- (1) Halgren, T. A.; Damm, W. Polarizable force fields. *Curr. Opin. Struct. Biol.* **2001**, *11* (2), 236–242.
- (2) Jing, Z.; Liu, C.; Cheng, S. Y.; Qi, R.; Walker, B. D.; Piquemal, J. P.; Ren, P. Polarizable Force Fields for Biomolecular Simulations: Recent Advances and Applications. *Annu. Rev. Biophys.* **2019**, *48*, 371–394.
- (3) Warshel, A.; Kato, M.; Pislakov, A. V. Polarizable Force Fields: History, Test Cases, and Prospects. *J. Chem. Theory Comput.* **2007**, *3* (6), 2034–2045.
- (4) Lamoureux, G.; MacKerell, A. D., Jr.; Roux, B. A simple polarizable model of water based on classical Drude oscillators. *J. Chem. Phys.* **2003**, *119* (10), 5185–5197.
- (5) Yu, H.; Hansson, T.; van Gunsteren, W. F. Development of a simple, self-consistent polarizable model for liquid water. *J. Chem. Phys.* **2003**, *118* (1), 221–234.
- (6) Rick, S. W.; Stuart, S. J.; Berne, B. J. Dynamical fluctuating charge force fields: Application to liquid water. *J. Chem. Phys.* **1994**, *101* (7), 6141–6156.
- (7) Burnham, C. J.; Li, J.; Xantheas, S. S.; Leslie, M. The parametrization of a Thole-type all-atom polarizable water model from first principles and its application to the study of water clusters ($n = 2$ –21) and the phonon spectrum of ice Ih. *J. Chem. Phys.* **1999**, *110* (9), 4566–4581.
- (8) Caldwell, J.; Dang, L. X.; Kollman, P. A. Implementation of nonadditive intermolecular potentials by use of molecular dynamics: development of a water-water potential and water-ion cluster interactions. *J. Am. Chem. Soc.* **1990**, *112* (25), 9144–9147.
- (9) Ren, P.; Ponder, J. W. Polarizable atomic multipole water model for molecular mechanics simulation. *J. Phys. Chem. B* **2003**, *107* (24), 5933–5947.
- (10) Wang, J.; Cieplak, P.; Li, J.; Wang, J.; Cai, Q.; Hsieh, M.; Lei, H.; Luo, R.; Duan, Y. Development of Polarizable Models for Molecular Mechanical Calculations II: Induced Dipole Models Significantly Improve Accuracy of Intermolecular Interaction Energies. *J. Phys. Chem. B* **2011**, *115* (12), 3100–3111.
- (11) Wei, H.; Qi, R.; Wang, J.; Cieplak, P.; Duan, Y.; Luo, R. Efficient formulation of polarizable Gaussian multipole electrostatics for biomolecular simulations. *J. Chem. Phys.* **2020**, *153* (11), 114116.
- (12) Tan, Y.-H.; Luo, R. Continuum treatment of electronic polarization effect. *J. Chem. Phys.* **2007**, *126* (9), 094103.
- (13) Tan, Y.-H.; Tan, C.; Wang, J.; Luo, R. Continuum Polarizable Force Field within the Poisson–Boltzmann Framework. *J. Phys. Chem. B* **2008**, *112* (25), 7675–7688.
- (14) Thole, B. T. Molecular polarizabilities calculated with a modified dipole interaction. *Chem. Phys.* **1981**, *59* (3), 341–350.
- (15) Lagardère, L.; Jolly, L.-H.; Lipparini, F.; Aviat, F.; Stamm, B.; Jing, Z. F.; Harger, M.; Torabifard, H.; Cisneros, G. A.; Schnieders, M. J.; et al. Tinker-HP: a massively parallel molecular dynamics package for multiscale simulations of large complex systems with advanced point dipole polarizable force fields. *Chem. Sci.* **2018**, *9* (4), 956–972.
- (16) Ponder, J. W.; Wu, C.; Ren, P.; Pande, V. S.; Chodera, J. D.; Schnieders, M. J.; Haque, I.; Mobley, D. L.; Lambrecht, D. S.; DiStasio, R. A., Jr.; et al. Current Status of the AMOEBA Polarizable Force Field. *J. Phys. Chem. B* **2010**, *114* (8), 2549–2564.
- (17) Cisneros, G. A.; Piquemal, J.-P.; Darden, T. A. Generalization of the Gaussian electrostatic model: Extension to arbitrary angular momentum, distributed multipoles, and speedup with reciprocal space methods. *J. Chem. Phys.* **2006**, *125* (18), 184101.
- (18) Piquemal, J. P.; Cisneros, G. A.; Reinhardt, P.; Gresh, N.; Darden, T. A. Towards a force field based on density fitting. *J. Chem. Phys.* **2006**, *124* (10), 104101.
- (19) Duke, R. E.; Starovoytov, O. N.; Piquemal, J. P.; Cisneros, G. A. GEM*: A Molecular Electronic Density-Based Force Field for Molecular Dynamics Simulations. *J. Chem. Theory Comput.* **2014**, *10* (4), 1361–1365.
- (20) Duke, R. E.; Cisneros, G. A. Ewald-based methods for Gaussian integral evaluation: application to a new parameterization of GEM. *J. Mol. Model.* **2019**, *25* (10), 307.
- (21) Naseem-Khan, S.; Piquemal, J.-P.; Cisneros, G. A. Improvement of the Gaussian Electrostatic Model by separate fitting of Coulomb and exchange-repulsion densities and implementation of a new dispersion term. *J. Chem. Phys.* **2021**, *155* (19), 194103.
- (22) Case, D. A.; Aktulga, H. M.; Belfon, K.; Cerutti, D. S.; Cisneros, G. A.; Cruzeiro, V. W. D.; Forouzes, N.; Giese, T. J.; Götz, A. W.; Gohlke, H.; et al. AmberTools. *J. Chem. Inf. Model.* **2023**, *63* (20), 6183–6191.
- (23) Elking, D.; Darden, T.; Woods, R. J. Gaussian induced dipole polarization model. *J. Comput. Chem.* **2007**, *28* (7), 1261–1274.
- (24) Guillot, B.; Guissani, Y. How to build a better pair potential for water. *J. Chem. Phys.* **2001**, *114* (15), 6720–6733.
- (25) Applequist, J.; Carl, J. R.; Fung, K.-K. Atom dipole interaction model for molecular polarizability. Application to polyatomic molecules and determination of atom polarizabilities. *J. Am. Chem. Soc.* **1972**, *94* (9), 2952–2960.
- (26) Cieplak, P.; Dupradeau, F.-Y.; Duan, Y.; Wang, J. Polarization effects in molecular mechanical force fields. *J. Phys.: Condens. Matter* **2009**, *21* (33), 333102.
- (27) Wei, H.; Cieplak, P.; Duan, Y.; Luo, R. Stress tensor and constant pressure simulation for polarizable Gaussian multipole model. *J. Chem. Phys.* **2022**, *156* (11), 114114.
- (28) Zhao, S.; Wei, H.; Cieplak, P.; Duan, Y.; Luo, R. PyRESP: A Program for Electrostatic Parameterizations of Additive and Induced Dipole Polarizable Force Fields. *J. Chem. Theory Comput.* **2022**, *18* (6), 3654–3670.
- (29) Zhao, S.; Wei, H.; Cieplak, P.; Duan, Y.; Luo, R. Accurate Reproduction of Quantum Mechanical Many-Body Interactions in Peptide Main-Chain Hydrogen-Bonding Oligomers by the Polarizable Gaussian Multipole Model. *J. Chem. Theory Comput.* **2022**, *18* (10), 6172–6188.
- (30) Huang, Z.; Zhao, S.; Cieplak, P.; Duan, Y.; Luo, R.; Wei, H. Optimal Scheme to Achieve Energy Conservation in Induced Dipole Models. *J. Chem. Theory Comput.* **2023**, *19* (15), 5047–5057.
- (31) Zhao, S.; Cieplak, P.; Duan, Y.; Luo, R. Transferability of the Electrostatic Parameters of the Polarizable Gaussian Multipole Model. *J. Chem. Theory Comput.* **2023**, *19* (3), 924–941.
- (32) Zhu, Q.; Wu, Y.; Zhao, S.; Cieplak, P.; Duan, Y.; Luo, R. Streamlining and Optimizing Strategies of Electrostatic Parameterization. *J. Chem. Theory Comput.* **2023**, *19* (18), 6353–6365.
- (33) Duan, Y.; Niu, T.; Wang, J.; Cieplak, P.; Luo, R. PCMRSP: A Method for Polarizable Force Field Parameter Development and

Transferability of the Polarizable Gaussian Multipole Models Across Multiple Solvents. *J. Chem. Theory Comput.* **2024**, *20* (7), 2820–2829.

(34) Zhao, S.; Cieplak, P.; Duan, Y.; Luo, R. Assessment of Amino Acid Electrostatic Parametrizations of the Polarizable Gaussian Multipole Model. *J. Chem. Theory Comput.* **2024**, *20* (5), 2098–2110.

(35) Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comput. Phys.* **1995**, *117* (1), 1–19.

(36) Darden, T.; York, D.; Pedersen, L. Particle mesh Ewald: An $N \log(N)$ method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98* (12), 10089–10092.

(37) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* **1995**, *103* (19), 8577–8593.

(38) Toukmaji, A.; Sagui, C.; Board, J.; Darden, T. Efficient particle-mesh Ewald based approach to fixed and induced dipolar interactions. *J. Chem. Phys.* **2000**, *113* (24), 10913–10927.

(39) Oh, K. J.; Deng, Y. An efficient parallel implementation of the smooth particle mesh Ewald method for molecular dynamics simulations. *Comput. Phys. Commun.* **2007**, *177* (5), 426–431.

(40) Martínez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M. PACKMOL: A package for building initial configurations for molecular dynamics simulations. *J. Comput. Chem.* **2009**, *30* (13), 2157–2164.

(41) *Amber 2024*; University of California: San Francisco, 2024.

(42) Gabriel, E.; Fagg, G. E.; Bosilca, G.; Angskun, T.; Dongarra, J. J.; Squyres, J. M.; Sahay, V.; Kambadur, P.; Barrett, B.; Lumsdaine, A.; et al. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface*; Kranzl Müller, D., Kacsuk, P., Dongarra, J., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2004; pp 97–104.

(43) Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.* **1982**, *76* (1), 637–649.

(44) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. C. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J. Comput. Phys.* **1977**, *23* (3), 327–341.

(45) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17* (3), 261–272.

(46) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; et al. Array programming with NumPy. *Nature* **2020**, *585* (7825), 357–362.

(47) Wang, J.; Cieplak, P.; Cai, Q.; Hsieh, M.-J.; Wang, J.; Duan, Y.; Luo, R. Development of Polarizable Models for Molecular Mechanical Calculations. 3. Polarizable Water Models Conforming to Thole Polarization Screening Schemes. *J. Phys. Chem. B* **2012**, *116* (28), 7999–8008.

(48) Douglass, D. C. Self-Diffusion and Velocity Correlation. *J. Chem. Phys.* **1961**, *35* (1), 81–90.

(49) Loncharich, R. J.; Brooks, B. R.; Pastor, R. W. Langevin dynamics of peptides: the frictional dependence of isomerization rates of N-acetylalanine-N'-methylamide. *Biopolymers* **1992**, *32* (5), 523–535.

(50) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, *81* (8), 3684–3690.



The banner features a collage of scientific images and text. On the left, a woman in a lab coat is shown. The central text reads 'CAS INSIGHTS™' in yellow, followed by 'EXPLORE THE INNOVATIONS SHAPING TOMORROW' in large white and blue letters. Below this, it says 'Discover the latest scientific research and trends with CAS Insights. Subscribe for email updates on new articles, reports, and webinars at the intersection of science and innovation.' A yellow button with the text 'Subscribe today' is positioned below the text. On the right, there are several smaller text boxes: 'Goldene—advancing new applications on the promise of graphene', 'Webinar: Emerging areas in biomaterials Reshaping medicine and human health', and 'Get new perspectives on the latest research in the field of science and innovation'. The bottom right corner features the CAS logo, which consists of the letters 'CAS' in white and a blue and yellow dot matrix pattern to the right. Below the logo, it says 'A division of the American Chemical Society'.