JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

Reconsidering Dispersion Potentials: Reduced Cutoffs in Mesh-Based Ewald Solvers Can Be Faster Than Truncation

Rolf E. Isele-Holder,*[,†] Wayne Mitchell,[†,‡] Jeff R. Hammond,[§] Axel Kohlmeyer, [∥] and Ahmed E. Ismail*[,†]

[†]Aachener Verfahrenstechnik: Molecular Simulations and Transformations and AICES Graduate School, RWTH Aachen University, Schinkelstraße 2, 52062 Aachen, Germany

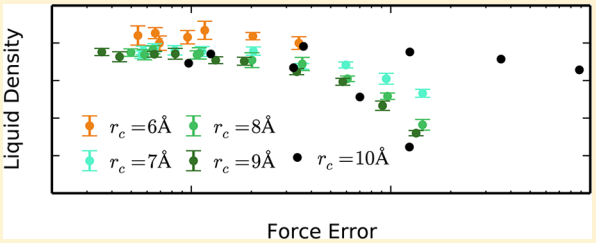
[‡]Loyola University, 6363 Saint Charles Avenue, New Orleans, Louisiana 70118, United States of America

[§]Leadership Computing Facility, Argonne National Laboratory, Argonne, Illinois 60439, United States of America

[∥]International Centre for Theoretical Physics, Strada Costiera 11, 34151 Trieste, Italy

Supporting Information

ABSTRACT: Long-range dispersion interactions have a critical influence on physical quantities in simulations of inhomogeneous systems. However, the perceived computational overhead of long-range solvers has until recently discouraged their implementation in molecular dynamics packages. Here, we demonstrate that reducing the cutoff radius for local interactions in the recently introduced particle−particle particle−mesh (PPPM) method for dispersion [Isele-Holder et al., *J. Chem. Phys.*, **2012**, *137*, 174107] can actually often be faster than truncating dispersion interactions. In addition, because all long-range dispersion interactions are incorporated, physical inaccuracies that arise from truncating the potential can be avoided. Simulations using PPPM or other mesh Ewald solvers for dispersion can provide results more accurately and more efficiently than simulations that truncate dispersion interactions. The use of mesh-based approaches for dispersion is now a viable alternative for all simulations containing dispersion interactions and not merely those where inhomogeneities were motivating factors for their use. We provide a set of parameters for the dispersion PPPM method using either *i***k** or analytic differentiation that we recommend for future use and demonstrate increased simulation efficiency by using the long-range dispersion solver in a series of performance tests on massively parallel computers.



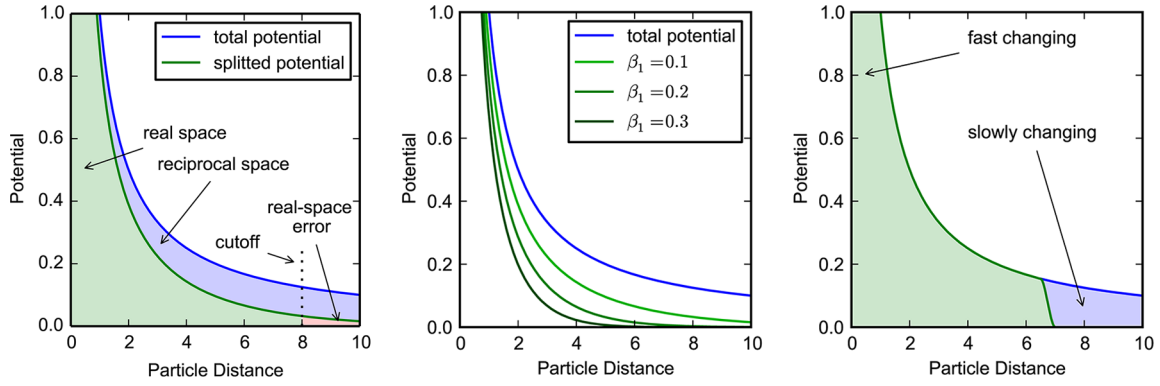
1. INTRODUCTION

Fluctuations in electronic wave functions lead to formation of instantaneous dipoles in atoms or molecules, which in turn induce dipoles in other molecules. When averaged over time, this interplay results in a weakly attractive force, known as dispersion, which acts between all kinds of materials. Because dispersion forces are purely attractive, they are especially relevant in interfacial science and, via the Hamaker constant, have influenced various other disciplines.[1] More recently, the treatment of dispersion interactions has drawn attention in quantum mechanics[2−5] and even computational mechanics at the continuum scale.[6]

Although dispersion interactions are formally short-ranged according to the mathematical definition, they can still have effects on the scale of tens of nanometers.[1] In molecular simulations, however, dispersion interactions have typically been truncated for distances beyond a cutoff of around 1 nm. For homogeneous systems, the error introduced by this truncation is usually acceptable because dispersion interactions decay quickly, errors in neglected forces partially cancel, and errors in energies and pressures can be corrected efficiently and accurately during the simulation.[7] For inhomogeneous systems, however, especially those involving interfaces, the conditions that justify the simple truncation of dispersion interactions are no longer met. To avoid the resulting errors, various methods for either correcting or explicitly computing long-range dispersion interactions have been developed.[8−18]

Proposed methods for incorporating the long-range effects of dispersion interactions include using extremely large cutoffs,[18] twin-range cutoffs,[12] schemes that perform corrections after[8,9,16] or during[11,13] the simulation, and explicit schemes using either traditional[15] or mesh-based[17,19] Ewald summation. Using extremely large cutoffs is simple and applicable with almost no restrictions, ensures highly accurate results, and requires little overhead in terms of code development. However, as the computing time needed to evaluate all pair interactions scales with $O(r_c^3)$, where r_c is the cutoff, this approach is extremely unattractive computationally. Correction-based methods are usually very efficient, but their accuracy and applicability are limited by the assumptions these methods are based on. Moreover, corrections that are applied during post-processing cannot influence the dynamics and behavior of the simulation. Traditional Ewald summation, similar to using large cutoffs, is accurate and versatile, but its $O(N^{3/2})$ scaling,[20]

Received: June 3, 2013
Published: November 5, 2013



(a) Ewald summation (b) Different Ewald parameters (c) Splitting with rRESPA

Figure 1. (a) Ewald summation splits a potential into terms that are solved in real or reciprocal space. The error in real space is related to the area under the potential curve beyond the cutoff. (b) The Ewald parameter influences the splitting function and can be used to tune the real-space error. (c) With rRESPA, the splitting of the potential can be performed alternatively. The fast changing part of the potential is computed more fequently than the slowly changing part.

where N is the number of particles, normally limits its applicability to simulations with only a few thousand particles.[21]

As we have recently shown,[17] mesh-based Ewald summation methods combine the advantages of the aforementioned methods: their accuracy can be tuned, they are applicable to arbitrary periodic systems, their computational overhead is small, and their $O(N \log N)$ scaling enables large-scale simulations. Moreover, mesh-based Ewald methods for dispersion interactions are similar to their Coulomb interactions counterparts. Consequently, adding dispersion particle−particle particle−mesh (PPPM) solvers to codes that already include either the PPPM[22] or the particle-mesh Ewald[23,24] (PME) method should be straightforward. Moreover, the existing body of literature for mesh-based Coulomb solvers can be adapted for dispersion solvers.[17,21−37]

Many of the advantages of mesh-based dispersion solvers are obvious from theoretical considerations, and their need has also been demonstrated in simulation studies.[8,10,13,38−40] Despite that, relatively limited development of these solvers[19,24] occurred until recent contributions by us[17] and by Wennberg et al.[41] To some extent, the development of these solvers has not been pushed forward earlier because homogeneous correction methods may be applicable and because adding mesh-based Ewald methods to a code requires significant effort although, as mentioned above, it is certainly manageable, especially if codes already have a mesh-based Coulomb solver. The most pressing criticism of mesh-based Ewald dispersion solvers is the belief that the required additional computations will impose an unacceptably high computational overhead.

We show that the assumption that mesh-based dispersion solvers inevitably lead to an increase in computing time is flawed and that simulations with the PPPM method for dispersion can actually be both faster and more accurate than using a truncated potential. The performance gain compared to simulations in which dispersion interactions are simply truncated emerges from the simulation cutoff becoming a tuning parameter for long-range dispersion solvers. Moreover, the use of long-range solvers leads to a different set of configurations that usually represent a given physical system more accurately. This can most readily be seen in thermodynamic and transport properties that depend on the density.

The combination of greater computational efficiency and physical accuracy means that the use of mesh-based Ewald methods is now a viable option for simulations of all systems including dispersion interactions. This is a significant expansion from our previous conclusions,[17] which recommended use of the dispersion PPPM solver for inhomogeneous systems.

In Section 2, we present the idea of using the cutoff as a tuning parameter. We also show limits of the applicability of the cutoff as a parameter for tuning the efficiency and provide a set of parameters for the PPPM method that lead to both accurate results and efficient simulations. These parameters have been used in large-scale performance tests that are described in Section 3. We show that simulations with the PPPM method for dispersion can outperform simulations in which dispersion interactions are truncated. We discuss our findings in Section 4 and summarize them in Section 5.

2. PERFORMANCE TUNING WITH FLEXIBLE CHOICE OF CUTOFFS

2.1. Theoretical Considerations in the Usage of Smaller Cutoffs. The principal idea behind Ewald methods (cf. Figure 1a) is to split up a given pair potential

$$u_p(r) = \frac{1}{r^p} = \frac{f_p(\beta_p, r)}{r^p} + \frac{1 - f_p(\beta_p, r)}{r^p} \quad (1)$$

where p is a positive integer; $f_p(\beta_p,r)$ is a splitting function, which is in principal arbitrary but usually derived from the Euler gamma function;[24] r is the distance between two particles; and β_p is the Ewald coefficient of the splitting function. The first term on the right-hand side of eq 1, which is also referred to as the real-space term, is solved by direct evaluation of the pair potential inside a chosen cutoff radius. The second term is called the reciprocal-space term and is solved using Fourier transforms for Ewald methods, and fast Fourier transforms (FFTs) for mesh-based Ewald methods.

The error introduced by truncating the real-space term is related to the area of the real-space potential beyond the cutoff. This error can be decreased by increasing either the cutoff radius r_c or the Ewald parameter β_p, as indicated in Figure 1b. This allows for more flexible choice of cutoffs, since for any r_c the Ewald parameter can be adjusted to achieve a given accuracy in the real-space computations. Because of the strong

dx.doi.org/10.1021/ct4004614 | J. Chem. Theory Comput. 2013, 9, 5412−5420

influence of the cutoff on the computation time, smaller cutoffs facilitated by the use of an Ewald method are a promising way to accelerate simulations.

While a desired real-space accuracy can be obtained for any choice of the cutoff radius r_c by adjusting the Ewald parameter, there are two different reasons why there is a lower bound for simulation cutoffs, even when using the Ewald method. First, as the cutoff radius decreases, the Ewald parameter must increase to achieve the same real-space accuracy.[17,30,35] Increasing the Ewald parameter, however, simultaneously reduces the accuracy of the reciprocal-space calculations, which can be corrected by increasing either the interpolation order or the number of grid points, both of which increase the computation time. Below a certain cutoff threshold, the Ewald parameter becomes so large that sufficient reciprocal-space accuracy cannot be achieved for any reasonable choice of the number of grid points and interpolation order.

The second reason for a lower limit of the cutoff arises because nonbonded potentials are usually composed of different terms. In such cases, the lower limit for the cutoff is determined by the truncation error of the slowest-decaying potential for which no long-range solver is applied. For example, consider the frequently used combination of Coulomb and Lennard-Jones (LJ) interactions

$$u_{nb,ij} = \frac{q_i q_j}{r_{ij}} - \frac{C_{ij}}{r_{ij}^6} + \frac{D_{ij}}{r_{ij}^{12}} \tag{2}$$

where q_i and q_j are the partial charges of particles i and j, r_{ij} is the distance between them, and $C_{ij} = 4\varepsilon_{ij}\sigma_{ij}^6$ and $D_{ij} = 4\varepsilon_{ij}\sigma_{ij}^{12}$ are LJ coefficients. The need for including a long-range solver for the Coulomb part of this potential is generally accepted. When using a mesh-based Ewald solver, for any cutoff radius r_c there exists an Ewald parameter β_1 that can achieve a desired accuracy for the real-space Coulomb term.

The error caused by truncating the LJ potential at r_c, however, cannot be compensated for if no long-range solver is used for the second and third terms in eq 2. In this case, the r^{-6} dispersion term in eq 2 controls the error. As a consequence the lower value for acceptable cutoffs is typically in the range from 10 Å to 14 Å depending on the desired accuracy and the selected force field in simulations of homogeneous systems or slightly inhomogeneous systems.[42−44] It is worth mentioning that the trend in molecular simulations has largely been moving toward using larger cutoffs.[18] The performance gain from using small cutoffs is counterbalanced by the truncation error of the r^{-6} term.

If an Ewald-based solver is also applied to the r^{-6} dispersion term in eq 2, the rapidly decaying repulsive term (here r^{-12}) becomes the slowest-decaying potential, and its truncation error determines the lower limit of r_c. This facilitates decreasing the computation time by using smaller cutoffs without loss of accuracy. As shown below, making the cutoff radius r_c a tuning parameter when using a PPPM dispersion solver can lead to a net increase in performance and accuracy compared to simulations in which dispersion interactions are simply truncated.

2.2. Numerical Tests with Smaller Cutoffs. We have performed a series of interfacial simulation with the OPLS-AA hexane[42] system to determine the lower limits for the cutoff when using a long-range dispersion solver. Except for the parameters for the long-range dispersion solver, simulations were performed as in our previous study.[17] For the dispersion interactions, we used real-space cutoffs of $r_c \in \{6,7,8,9\}$ Å. The uniform grid spacings and ranges of Ewald parameters used in these simulations is given in Table 1. The cutoff for the repulsive part of the LJ potential was equal to the cutoff used for the dispersion part.

Table 1. Parameters Used for PPPM for Dispersion Interactions in Simulations for Finding Lower Values for the Simulation Cutoff

r_c [Å]	h_6 [Å]	β_6 [Å^{-1}]
6	50/96	0.48−0.6
7	50/64	0.34−0.48
8	50/48	0.25−0.42
9	50/40	0.2−0.37

To analyze the results each trajectory has been subdivided into five parts. For each of these parts, density profiles have been fitted and liquid densities determined accordingly. The averages of the computed densities are reported in Figure 2.

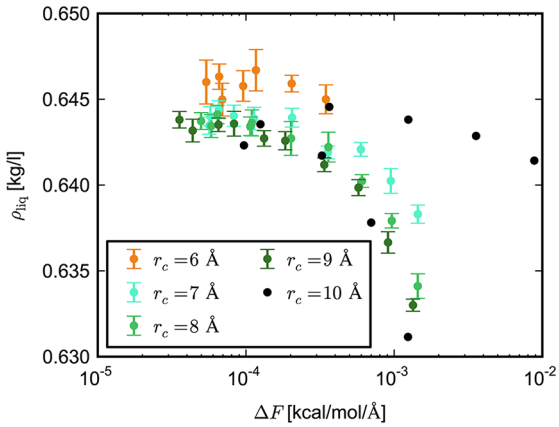


Figure 2. Simulated densities with different cutoffs r_c for the LJ potential and different Ewald parameters β_6 as a function of the measured error in the forces. Data for r_c = 10 Å was taken from ref 17. Results for r_c = 6 Å are quantitatively different from those obtained with larger cutoffs.

The plotted error bars are twice the computed standard error in the mean in each direction. Results from our previous study[17] with a cutoff of 10 Å are also included in the figure. Simulated densities are given as a function of the average error in the forces. This error was determined from a single, well-equilibrated configuration via

$$\Delta F_{\text{disp}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{F}_{i,\text{disp}}^{\text{PPPM}} - \mathbf{F}_{i,\text{disp}}^{\text{exact}})^2} \tag{3}$$

where N is the total number of particles, $\mathbf{F}_{i,\text{disp}}^{\text{PPPM}}$ is the force on particle i caused by the dispersion potential computed with the PPPM Ewald method with the used parameters, and $\mathbf{F}_{i,\text{disp}}^{\text{exact}}$ is the "exact" force on particle i caused by the dispersion potential. $\mathbf{F}_{i,\text{disp}}^{\text{exact}}$ was computed with the PPPM method for dispersion with a large real-space cutoff and very fine grid spacing such that the obtained accuracy is very high. The cutoff and grid spacing were chosen such that the error in the reference computations was at most 5×10^{-8} kcal mol^{-1} Å^{-1} according to error estimates in ref 17.

dx.doi.org/10.1021/ct4004614 | J. Chem. Theory Comput. 2013, 9, 5412−5420

Figure 2 shows that, as the accuracy of computed dispersion forces increases, the simulated densities reach a plateau for each of the chosen cutoff radii. For cutoffs $r_c \in \{7,8,9,10\}$ Å, this plateau is located at approximately 0.643 kg/L. For a cutoff radius of r_c = 6 Å, however, the plateau is slightly higher at approximately 0.646 kg/L. The increased simulated density arises because the particles feel significantly less repulsion for this small cutoff, indicating that errors caused by truncating the repulsive part of the LJ potential begin to affect the behavior of the system. For cutoffs $r_c \geq 7$ Å, however, simulated densities are indistinguishable from results obtained in simulations with larger cutoff radii in the plateau region in Figure 2.

Errors in the forces ΔF_{rep} that arise from truncating the repulsive potential have been determined similar to eq 3. For a cutoff of r_c = 7 Å the error is approximately $\Delta F_{rep} \approx 0.00015$. For a cutoff of r_c = 6 Å, the measured error is approximately $\Delta F_{rep} \approx 0.0009$. As the minimum possible cutoff is determined from the maximum LJ diameter (σ_{CC} = 3.5 Å in these simulations), we deduce that cutoffs down to twice the maximum LJ diameter are applicable when using long-range dispersion solvers without influencing the results because of truncating the repulsion term.

2.3. Parameter Selection for Optimal Accuracy and Efficiency. In this section, we present a set of parameters for PPPM for disperson and briefly review the employed methods for accelerating the simulations with which increased accuracy and efficiency can be obtained. The proof that the used techniques and parameters provide accurate results is given in this section, whereas the performance benefit is demonstrated in Section 3.

As maximum performance for the reciprocal-space computations can be obtained using *i*k or analytic (ad) differentiation depending on the simulations settings,[30] we provide parameters for both options. Self-force correction has been applied to Coulomb and dispersion forces in simulations with the ad differentiation.[34] To decrease the effort for the real-space computations, we used the double-precision extraction method of Wolff and Rudd.[45] Moreover, we have employed the reversible reference system propagator algorithm[46] (rRESPA), which uses a different force splitting, which requires less frequent computation of the reciprocal-space forces (cf. Figure 1c).[27] Usage of rRESPA reduces the effort for the reciprocal-space calculations by roughly n_r, the ratio of how often the fast-changing part of the potential is evaluated compared to the slowly changing part. We carried out rRESPA simulations with n_r = 4. The fast-changing part of the potential was smoothly shifted to zero between 6.5 Å and 7.0 Å and evaluated every femtosecond.

PPPM parameters for Coulomb and dispersion interactions were chosen such that the simulation results are not influenced by inaccuracies of the PPPM method. Parameters for the Coulomb interactions were generated by applying LAMMPS routines, based on error estimates summarized in ref 35, to a bulk sample of water with a desired accuracy of $\Delta F \approx 0.03$. Dispersion parameters were chosen such that, for the hexane system, the real-space error was $\Delta F \approx 0.0002$ kcal mol^{-1} Å^{-1}, which is the threshold above which the real-space error influences the simulation results. This can be seen from Figure 2, considering that with the PPPM parameters used in Section 2.2 the real-space error is approximately equal to the total error at the point where simulated densities start to decrease. The reciprocal-space error was $\Delta F \approx 0.004$ kcal mol^{-1} Å^{-1}. This value for the required reciprocal-space accuracy for dispersion interactions was taken from our previous study[17] and is in good agreement with the recent findings by Wennberg et al.[41] As previously observed,[17] for the PPPM dispersion solver, the real-space accuracy needs to be higher than the reciprocal-space accuracy. The resulting parameters are summarized in Table 2; more information on the choice of simulation parameters is given in the Supporting Information.

Table 2. Ewald Parameter and Grid Spacing for PPPM Validation Simulations[a]

	β_1 [Å^{-1}]		h_1 [Å]		β_6 [Å^{-1}]	h_6 [Å]	
r_c [Å]	*i*k	ad	*i*k	ad	*i*k and ad	*i*k	ad
7	0.373	0.375	1.3889	1.25	0.40	2.0	1.6667
8	0.328	0.324	1.5625	1.3889	0.33	2.5	2.0833
9	0.287	0.294	1.8519	1.3889	0.28	3.125	3.125
10	0.257	0.259	2.0833	1.6667	0.24	6.25	6.25

[a]The interpolation order in the simulations was P = 5.

The parameters have been validated in a series of interfacial simulations of hexane, which was modeled with the OPLS-AA force field,[42] and SPC/E[47] water. If not otherwise explicitly stated, simulations were performed as described in our previous work.[17] Production runs for results reported in this section were run for 5 ns. In addition to liquid densities, surface tensions γ were determined using the approach of Kirkwood and Buff.[48]

Results of the validation simulations are given in Table 3. For each parameter combination, simulation results are within statistical uncertainties equal to results from our previous work.[17] Hence, inaccuracies introduced by PPPM parameters or the applied acceleration methods do not alter the simulation results. In contrast, simulations not using a long-range dispersion solver provide results that are quantitatively different. PPPM coefficients for dispersion interactions in Table 2 were successfully transferred from hexane to SPC/E water. They are expected to work in similar systems and are recommended for future use.

3. PERFORMANCE SIMULATIONS

We have performed a series of performance runs on the Vesta Blue Gene/Q system at the Argonne Leadership Computing Facility. To show that our findings are transferable to other architecture types, we performed similar performance tests on the SuperMUC Sandy Bridge-EP processors at the Leibniz Rechenzentrum in Garching, Germany, which are reported in the Supporting Information. Performance simulations have been performed with the 22Mar13 version of LAMMPS; compilation information for the machines used is provided in the Supporting Information. The parallelization strategies for the parts of the code used here can be found in ref 27 and ref 49.

3.1. Simulation Setup. We have chosen SPC/E water as the test case for our performance comparisons. The model is widely known and used in the molecular simulation community. Furthermore, it requires long-range Coulomb solvers, while also allowing use of long-range dispersion solvers. The performance of the latter is dependent on the chosen mixing rule.[15] To examine this effect simulations were run with routines that compute long-range dispersion interactions for

Table 3. Simulated Liquid Densities and Surface Tensions for Hexane and SPC/E

		hexane				SPC/E			
		$n_r = 1$		$n_r = 4$		$n_r = 1$		$n_r = 4$	
diff	r_c [Å]	ρ_{liq} [kg/L]	γ [mN/m]	ρ_{liq} [kg/L]	γ [mN/m]	ρ_{liq} [kg/L]	γ [mN/m]	ρ_{liq} [kg/L]	γ [mN/m]
ad	7	0.6427 (5)	16.40 (60)	0.6431 (8)	16.18 (60)	0.9962 (3)	60.28 (100)	0.9955(4)	60.46 (90)
	8	0.6428 (7)	16.57 (70)	0.6425 (6)	16.52 (60)	0.9958 (3)	60.94 (100)	0.9952 (2)	59.69 (90)
	9	0.6428 (3)	16.22 (60)	0.6423 (3)	16.18 (60)	0.9956 (3)	60.78 (90)	0.9952 (3)	60.38 (90)
	10	0.6443 (7)	16.23 (50)	0.6437 (5)	16.12 (50)	0.9954 (3)	60.05 (90)	0.9962 (2)	58.41 (90)
ik	7	0.6424 (7)	16.04 (50)	0.6428 (3)	16.27 (50)	0.9970 (3)	59.60 (90)	0.9962 (3)	59.49 (90)
	8	0.6432 (3)	16.30 (50)	0.6426 (6)	16.09 (50)	0.9964 (3)	59.82 (90)	0.9956 (3)	60.68 (80)
	9	0.6430 (8)	15.99 (60)	0.6423 (3)	16.18 (60)	0.9961 (4)	60.15 (80)	0.9955 (4)	60.20 (90)
	10	0.6446 (4)	16.08 (60)	0.6438 (6)	16.26 (50)	0.9964 (4)	57.96 (80)	0.9956 (5)	58.71 (90)
	10[a]	0.6434	16.41(50)			0.9965	60.72(90)		
	10[a,b]	0.6058	12.42(50)[c]			0.9882	59.86(100)[c]		
	12[a,b]	0.6251	13.96(50)[c]			0.9918	60.10(100)[c]		

[a]Taken from our previous work[17] with different PPPM settings. [b]With a truncated potential. [c]Reported value includes tail correction.

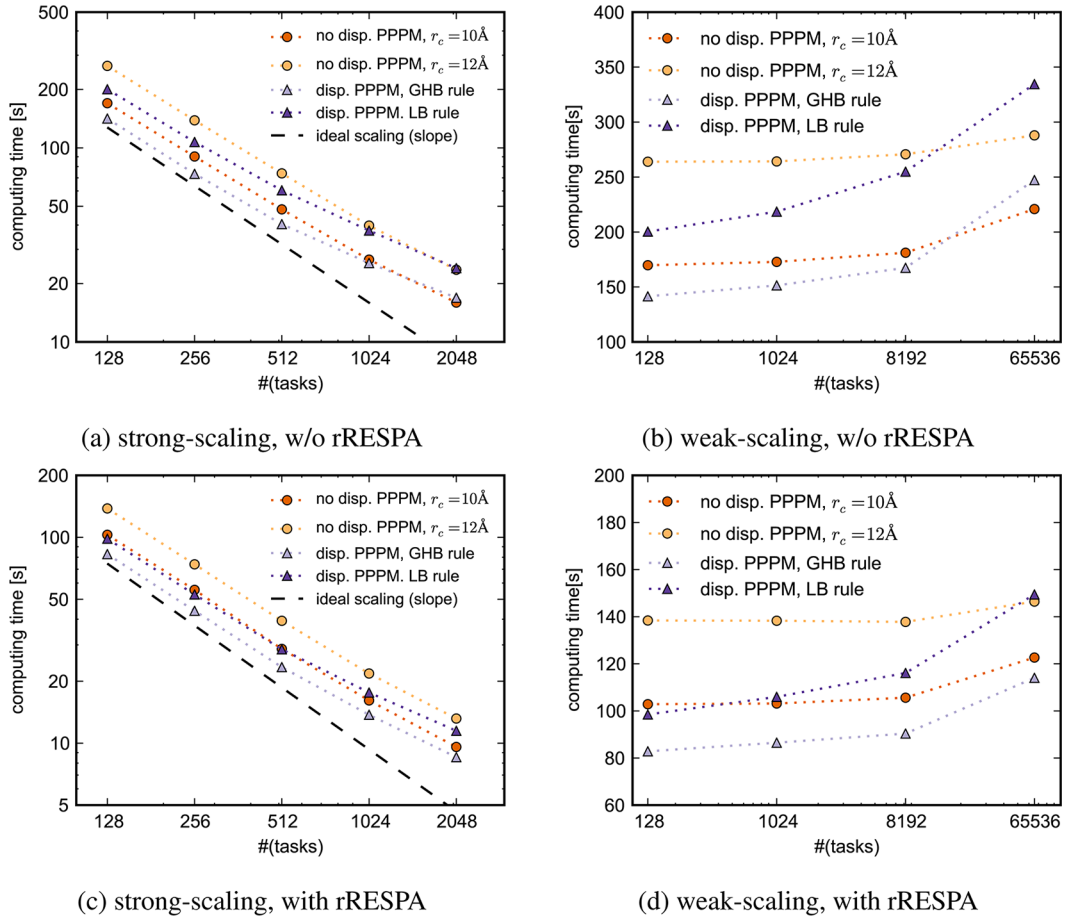


Figure 3. Performance of simulations that provided fastest computations for a given number of particles and CPUs.

the popular combinations of the Berthelot[50] mixing rule for energies

$$\varepsilon_{ij} = \sqrt{\varepsilon_{ii}\varepsilon_{jj}} \tag{4}$$

with the Lorentz[51] and Good−Hope[52] mixing rules for the LJ σ-parameters:

$$\sigma_{ij} = \frac{\sigma_{ii} + \sigma_{jj}}{2}, \quad \text{(Lorentz)} \tag{5}$$

$$\sigma_{ij} = \sqrt{\sigma_{ii}\sigma_{jj}}, \quad \text{(Good − Hope)} \tag{6}$$

Combinations of Good−Hope and Berthelot (GHB) rules are computationally favorable because the dispersion coefficients C_{ij} in eq 2 follow a geometric mixing rule. Performance results reported below for this mixing rule are representative for all mixing rules that provide geometric mixing of the dispersion coefficients.[53,54] In contrast, use of the Lorentz−Berthelot (LB) mixing rule requires multiple reciprocal-space computations. A more detailed description of the influence of the mixing rule, current strategies for treating different mixing rules, and a

dx.doi.org/10.1021/ct4004614 | J. Chem. Theory Comput. 2013, 9, 5412−5420

strategy to reduce the computational overhead and to enable application of long-range dispersion solvers to systems that do not follow any mixing rule is given in Appendix A.

To create an initial starting configuration, a small system of 666 SPC/E molecules has been equilibrated in a cubic box with a volume of 20191 Å^3, which corresponds to a density of approximately 1 kg/L. This system has been equilibrated for 20 ps in the NVT ensemble at 300 K. Configurations for the performance simulations were obtained by simply replicating the equilibrated system. Performance simulations were run for 1 ps.

We have performed a series of simulations with the PPPM method for dispersion and with truncated potentials. Simulations using truncated potentials were performed with cutoff radii of 10 and 12 Å, as these are typically used values for bulk phase simulations. Simulations with the PPPM dispersion method were performed with cutoff radii of 7, 8, 9, and 10 Å. As shown in Section 2.3, simulation results are independent of cutoffs, provided the remaining parameters are chosen appropriately. These cutoffs can therefore be considered as tuning parameters. All comparisons are between a simulation using the PPPM method for dispersion to a simulation using a truncated potential, with all other features and settings kept the same.

We performed weak-scaling and strong-scaling tests. In the weak-scaling tests, the average number of particles per task (where the number of tasks is the number of MPI tasks times the number of OpenMP threads per MPI task) was kept constant at 999. The number of tasks in the weak-scaling test were 128, 1024, 8192, and 65536, corresponding to 127 872, 1 022 976, 8 183 808, and 65 470 464 particles. In strong-scaling tests, the overall number of particles was kept constant at 127 872. The number of tasks were 128, 256, 512, 1024, and 2048.

We also examined the effect of maximizing the performance with hybrid parallelization. We have varied the number of OpenMP threads per MPI task between 2, 4, and 8 for all simulations. Additionally, we have performed all simulations with and without rRESPA. rRESPA settings are as reported in Section 2.3.

PPPM parameters for Coulomb interactions were selected as described in Section 2.3. Parameters for the PPPM method for dispersion interactions are almost identical to those in Table 2. Slightly different values for the grid spacing were chosen because the number of grid points in each direction is restricted to values that can be factored by 2, 3, and 5. Analytic differentiation has been used in the performance simulations. Evaluation of the pair potentials was accelerated using tabulation. Sample input scripts for the different simulation settings are given in the Supporting Information.

3.2. Results. In this section, we report simulation times for settings that provided maximum performance for a given system and number of tasks. For simulations run without PPPM for dispersion, results for r_c = 10 Å and r_c = 12 Å are reported because these are the most frequently observed cutoffs for dispersion terms in commonly used force fields. For simulations with PPPM for dispersion, the cutoff is a tuning parameter and we therefore report only the values that provide the fastest computations. Figure 3 shows the total simulation time of the performance runs as a function of the number of tasks. To allow for a better quantitative comparison, the results of all performance runs are tabulated in the Supporting Information.

In simulations without rRESPA and with a fixed number of particles (cf. Figure 3a), the fastest simulations were obtained when the PPPM method for dispersion was used with the GHB mixing rule. Shorter simulation times were only obtained without the long-range dispersion solver for systems with around 60 particles per task. Simulations with PPPM for dispersion and the LB mixing rule usually outperformed simulations without the PPPM method for dispersion and a cutoff of 12 Å. Similar behavior is observed in tests with an increasing number of particles, as shown in Figure 3b.

In simulations with rRESPA, the best performance in all test cases was achieved when using the PPPM method for dispersion with the GHB mixing rule. For the lowest number of particles or the highest load of particles per core, the speed-up compared to simulations with a truncated potential and a cutoff of 12 Å is more than 40%. Even the unfavorable LB mixing rule often provided a better performance than simulations with a truncated potential cutoff of 10 Å. The PPPM method for dispersion with the LB mixing rule was outperformed by the simulations using a truncated potential and a cutoff of 12 Å only in the simulations of the largest system.

When moving toward larger numbers of particles or a larger number of processors, larger cutoffs become more favorable, the PPPM method for dispersion becomes computationally less advantageous, and an increased number of OpenMP threads provides faster simulations (cf. Tables in Supporting Information). These observations are related to the limitations of parallel FFTs, which require data to be contiguous in one dimension and thus can be parallelized only in two dimensions. They also require a redistribution of data before, in between, and after individual parallelized transforms, leading to increased communication with more processors. For very large problems and a very large number of processors, it is faster to do the reciprocal-space calculation with a subset of communicating processors via hybrid MPI−OpenMP parallelization or on a separate partition of processors. At the very extreme scale, a multigrid approach offers better scaling behavior in exchange for a larger prefactor but uses the same kind of short-range versus long-range split in the calculation of the interactions.[55,56] We anticipate a similar performance benefit by including long-range dispersion interactions for these solvers with reduced short-ranged cutoffs, too. As PPPM for dispersion requires additional FFTs and the number of grid points for the Coulomb interactions is larger when using smaller cutoffs, the benefits and shortcomings of the FFTs dominate for simulations with the long-range dispersion solver; exploring options to reduce the communication overhead for these cases is thus of importance.

4. DISCUSSION

Whether the PPPM method for dispersion can be used to decrease the simulation time depends on multiple factors, including the system size, the level of parallelization, the employed mixing rule, the machine used for the simulations, as well as what cutoff would be considered appropriate in simulations using a truncated potential. Moreover, the use of rRESPA has an even stronger influence on the results. Therefore drawing universally valid conclusions is difficult, yet it is possible to state that a performance gain using a PPPM method for dispersion was achieved in most of the test cases described here. This includes cases using the computationally unfavorable LB mixing rule. Moreover, even when using a long-

range dispersion solver did not lead to a better performance, the disadvantage was not unaffordably large.

When comparing performance, we note that simulations with the PPPM method for dispersion provide better accuracy than simulations without a long-range solver. Especially for strongly inhomogeneous systems, much larger cutoffs (e.g., r_c = 23 Å as suggested in ref 18.) are required to provide accurate results. Such large cutoffs are much more expensive than any of the results reported in this study. Hence, the PPPM method for dispersion is a method that substantially increases the accuracy with gains or, at worst, minor losses in efficiency, depending upon the simulation settings. Its application can therefore be advantageous even when the physics of a system do not seem to demand the use of a long-range dispersion solver.

Additional improvements in the performance of the PPPM method for dispersion can be obtained, although they have not been used here. As shown in Appendix A, simulation times when using the LB mixing rule can often be reduced through an alternate splitting technique. Moreover, interlacing,[32,33] a technique for further increasing the efficiency of reciprocal-space computations, has not been employed. This method, similar to any other method that improves reciprocal-space efficiency, will favor the PPPM method for dispersion even more. We note that our approach is also applicable to other dispersion potentials, such as force fields that use an r^{-7} term. However, for force fields such as the Born potential[57,58] that contain multiple dispersion terms (such as linear combinations of r^{-6}, r^{-8}, and r^{-10}), the technique will become less advantageous because each term requires a separate reciprocal-space computation.

For a given simulation, one must consider not only how including the long-range dispersion solvers affects the efficiency but also how the physical behavior of simulations changes and whether this change is desirable. The physical nature of dispersion is not a simple decay to zero beyond the cutoff. However, many force fields truncate dispersion interactions for computational reasons. The parameters in those force fields are thus partly designed to compensate for errors caused by truncation of the dispersion potential at the cutoff. Applying a long-range dispersion solver thus usually means that the force field is used differently than intended.

Whether this alternate usage of the force field is acceptable or even desired depends on the simulation and the force field. As described in Section 1, the need for long-range dispersion interactions is generally accepted in strongly inhomogeneous simulations, which is why the deviation from the parametrization settings of the force field is of interest here. For force fields whose dispersion coefficients were not determined from fitting but, for example, from *ab initio* calculations, using long-range dispersion solvers is also beneficial. For force fields that have been parametrized either using a large cutoff (such as GROMOS96[44] with a cutoff of 14 Å) or small cutoffs with long-range corrections, such as OPLS-AA[42] or TraPPE,[43] the difference in obtained physical results might be sufficiently small that the decision whether to use a PPPM method for dispersion can be made solely by focusing on the performance. These considerations are in agreement with recent results by Wennberg et al.[41] For force fields that have been designed to be used with small cutoffs and without long-range corrections, the differences in simulation results are expected to be relatively large. A long-range solver may not be advantageous under such conditions.

Optimal results will be achieved, however, when the solver is used together with a force field that has been parametrized taking long-range dispersion interactions into account. As the use of PPPM for dispersion facilitates simulations that are more accurate and more efficient, we suggest that future force-field development should take long-range dispersion interactions into account alongside long-range Coulomb interactions.

5. CONCLUSIONS

Applying long-range dispersion solvers transforms the potential cutoff into a parameter for tuning the efficiency of a simulation. A lower limit for the cutoff is determined by the slowest-decaying potential for which no long-range solver is applied. We have shown that, for LJ potentials, cutoffs as small as twice the maximum LJ diameter are possible without altering the simulation results. A study of the influence of errors in the forces on simulation results was used to find parameters for the PPPM method for dispersion that provides more accurate results and increased efficiency compared to our previous recommendations.[17] The new set of parameters, summarized in the last three columns of Table 2, is recommended for future use.

We have also compared the performance of simulations with and without the PPPM method for dispersion. Because performance with or without the long-range dispersion solver depends strongly on several factors, drawing universally valid conclusions is difficult. We have shown, however, that simulations with the PPPM method for dispersion were faster than those not using it in various simulation settings. For scenarios where using the PPPM method for dispersion was optimal, the efficiency gain was more than 40%. Using a PPPM dispersion solver thus not only provides better accuracy but increases efficiency as well. Even where PPPM is not faster than a truncated potential, the performance overhead is so small that the improved accuracy of the simulation remains a sufficient justification for its use. Because long-range dispersion solvers can now provide better accuracy and better performance, we believe that future work in the area of force-field development should give serious consideration to long-range dispersion effects, just as they now handle long-range Coulomb interactions.

APPENDIX A: METHODS FOR SPLITTING DISPERSION COEFFICIENTS

The attractive part of the LJ potential is typically formulated as

$$u_6 = \frac{C_{ij}}{r^6} = \frac{4\varepsilon_{ij}\sigma_{ij}^6}{r^6} \tag{7}$$

where C_{ij}, ε_{ij}, and σ_{ij} are the dispersion and LJ coefficients. The mixing for the parameters is computationally optimal when eq 7 can be rewritten as

$$u_6 = \frac{a_i a_j}{r^6} \tag{8}$$

which is the case when $C_{ij} = (C_{ii}C_{jj})^{1/2} = a_i a_j$. This geometric mixing is fulfilled for several mixing rules,[52−54] but for the popular LB mixing rule, for example, the mixing coefficients are determined from eq 4 and eq 5, resulting in

$$C_{ij} = 4\sqrt{\varepsilon_{ii}\varepsilon_{jj}}\left(\frac{\sigma_{ii}+\sigma_{jj}}{2}\right)^6 = \frac{1}{16}\sqrt{\varepsilon_{ii}\varepsilon_{jj}}(\sigma_{ii}^6\sigma_{jj}^0 + 6\sigma_{ii}^5\sigma_{jj}^1 + 15\sigma_{ii}^4\sigma_{jj}^2 + 20\sigma_{ii}^3\sigma_{jj}^3 + 15\sigma_{ii}^2\sigma_{jj}^4 + 6\sigma_{ii}^1\sigma_{jj}^5 + \sigma_{ii}^0\sigma_{jj}^6)$$
$$= a_{i,0}a_{j,6} + a_{i,1}a_{j,5} + a_{i,2}a_{j,4} + a_{i,3}a_{j,3} + a_{i,4}a_{j,2} + a_{i,5}a_{j,1} + a_{i,6}a_{j,0}. \quad (9)$$

This splitting can be more generally written as a matrix equation:

$$\mathbf{C} = \mathbf{ADPA}^T \quad (10)$$

where $\mathbf{C}$ is an $n \times n$ matrix of the dispersion coefficients, $\mathbf{A}$ is a $n \times k$ matrix with transpose $\mathbf{A}^T$ that contains the a_i, $\mathbf{D}$ is a diagonal $k \times k$ matrix, and $\mathbf{P}$ is a $k \times k$ permutation matrix. Reciprocal-space computations are performed k times for this splitting. Eq 10 is generally applicable for splitting the dispersion coefficients in the framework of Ewald summation and is not restricted to the LB rule. When the LB mixing rule is used and dispersion coefficients are split with eq 9, k is equal to 7, the diagonal matrix $\mathbf{D}$ is the identity matrix, and $\mathbf{P}$ is antidiagonal.

The computation time can be optimized by performing the matrix factorization in eq 10 so that k is minimized; however, we are unaware of a general solution to this optimization problem. Thus, instead of solving eq 10, we propose to split dispersion coefficients with

$$C_{i,j} = d_0 a_{i,0}a_{j,0} + d_1 a_{i,1}a_{j,1} + \ldots + d_k a_{i,k}a_{j,k} \quad (11)$$

which can be written in a matrix form

$$\mathbf{C} = \mathbf{ADA}^\mathbf{T} \quad (12)$$

which is eq 10 when $\mathbf{P}$ is the identity matrix. Because $\mathbf{C}$ is symmetric, the matrix factorization in eq 12, which is equivalent to an eigenvalue decomposition, is always possible such that $k = n$. Moreover, if $\mathrm{rank}(\mathbf{C}) < n$, the decomposition can be performed such that $k = \mathrm{rank}(\mathbf{C})$.

The functional form of eq 11 was chosen such that it can easily be used in Ewald solvers. The a coefficients are assigned to the particles, whereas the d coefficients are simply multiplied by the Green's function used for the mesh-based solver. An advantage of this splitting method is that if the LB mixing rule is used, the dispersion potential is split into fewer terms if $\mathrm{rank}(\mathbf{C}) < 7$. This is equivalent to requiring that the number of different LJ diameters be less than 7, which is true in many simulations. In such cases, the overhead for the reciprocal-space computations is reduced. This splitting technique becomes unfavorable, though, if the number of different LJ diameters is greater than 7. This splitting method can also be applied to systems in which dispersion coefficients deviate from or do not follow any standard mixing rule.[59−62]

We would like to note that simulations with the LB mixing rule in Section 3 were performed with the splitting in eq 9, because this splitting provides performance independently of $\mathrm{rank}(\mathbf{C})$. This was realized by simply using routines that are generally applicable for the LB mixing rule in our simulations. Performance results for the LB mixing rule in Section 3 are thus also valid if the number of different LJ diameters is large.

■ ASSOCIATED CONTENT

Supporting Information

Scripts for compiling LAMMPS on the used computer architectures. Sample input scripts for the performance simulations plus an overview over all results of the performance tests. More information on the used PPPM parameters. This material is available free of charge via the Internet at http://pubs.acs.org/.

■ AUTHOR INFORMATION

Corresponding Authors

*E-mail: isele@aices.rwth-aachen.de.
*E-mail: aei@alum.mit.edu.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank Fabian Key for assistance in setting up some of the pilot simulations used in this paper, Steve Plimpton and Paul Crozier for support in integrating the PPPM for dispersion into the main LAMMPS release, and Pieter in 't Veld and David Cerutti for helpful discussions before starting this project. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357. Computer resources for this project have also been provided by the Gauss Centre for Supercomputing/Leibniz Supercomputing Centre under Grant No. PR89KA. Financial support for REI from the Deutsche Forschungsgemeinschaft (German Research Foundation) through Grant No. GSC 111 is gratefully acknowledged.

■ REFERENCES

(1) Israelachvili, J. *Intermolecular and Surface Forces*, 3rd ed.; Elsevier: Amsterdam, 2011.
(2) Grimme, S. *J. Comput. Chem.* **2006**, *27*, 1787−1799.
(3) Marom, N.; Tkatchenko, A.; Scheffler, M. *J. Chem. Theory Comput.* **2009**, *6*, 81−90.
(4) Tkatchenko, A.; Scheffler, M. *Phys. Rev. Lett.* **2009**, *102*, 073005.
(5) Tkatchenko, A.; DiStasio, R.; Head-Gordon, M.; Scheffler, M. *J. Chem. Phys.* **2009**, *131*, 094106.
(6) Sauer, R.; Wriggers, P. *Comput. Method. Appl. M.* **2009**, *198*, 3871−3883.
(7) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Oxford University Press: Oxford, 1987.
(8) Chapela, G.; Saville, G.; Thompson, S.; Rowlinson, J. *J. Chem. Soc. Faraday Trans. 2* **1977**, *73*, 1133−1144.
(9) Blokhuis, E. M.; Bedeaux, D.; Holcomb, C. D.; Zollweg, J. A. *Mol. Phys.* **1995**, *85*, 665−669.
(10) Guo, M.; Peng, D.-Y.; Lu, B. C.-Y. *Fluid Phase Equilib.* **1997**, *130*, 19−30.
(11) Mecke, M.; Winkelmann, J.; Fischer, J. *J. Chem. Phys.* **1997**, *107*, 9264−9270.
(12) Lagüe, P.; Pastor, R. W.; Brooks, B. R. *J. Phys. Chem. B* **2004**, *108*, 363−368.
(13) Janeček, J. *J. Phys. Chem. B* **2006**, *110*, 6264−6269.
(14) López-Lemus, J.; Romero-Bastida, M.; Darden, T. A.; Alejandre, J. *Mol. Phys.* **2006**, *104*, 2413−2421.
(15) in 't Veld, P. J.; Ismail, A. E.; Grest, G. S. *J. Chem. Phys.* **2007**, *127*, 144711.
(16) Shirts, M. R.; Mobley, D. L.; Chodera, J. D.; Pande, V. S. *J. Phys. Chem. B* **2007**, *111*, 13052−13063.
(17) Isele-Holder, R. E.; Mitchell, W.; Ismail, A. E. *J. Chem. Phys.* **2012**, *137*, 174107.

(18) Zubillaga, R. A.; Labastida, A.; Cruz, B.; Martínez, J. C.; Sánchez, E.; Alejandre, J. *J. Chem. Theory Comput.* **2013**, *9*, 1611−1615.
(19) Shi, B.; Sinha, S.; Dhir, V. K. *J. Chem. Phys.* **2006**, *124*, 204715.
(20) Perram, J. W.; Petersen, H. G.; de Leeuw, S. W. *Mol. Phys.* **1988**, *65*, 875−893.
(21) Pollock, E. L.; Glosli, J. *Comput. Phys. Commun.* **1996**, *95*, 93−110.
(22) Hockney, R.; Eastwood, J. *Computer Simulations Using Particles*; McGraw-Hill Inc.: New York, 1988.
(23) Darden, T.; York, D.; Pederson, L. *J. Chem. Phys.* **1993**, *98*, 10089−10092.
(24) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. *J. Chem. Phys.* **1995**, *103*, 8577−8593.
(25) Kolafa, J.; Perram, J. W. *Mol. Simulat.* **1992**, *9*, 351−368.
(26) Petersen, H. G. *J. Chem. Phys.* **1995**, *103*, 3668−3679.
(27) Plimpton, S.; Pollock, R.; Stevens, M. *Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing* **1997**, 8−21.
(28) Darden, T. A.; Toukmaji, A.; Pedersen, L. G. *J. Chim. Phys.* **1997**, *94*, 1346−1364.
(29) Deserno, M.; Holm, C. *J. Chem. Phys.* **1998**, *109*, 7678−7693.
(30) Deserno, M.; Holm, C. *J. Chem. Phys.* **1998**, *109*, 7694−7702.
(31) Stern, H. A.; Calkins, K. G. *J. Chem. Phys.* **2008**, *128*, 214106.
(32) Cerutti, D. S.; Duke, R. E.; Darden, T. A.; Lybrand, T. P. *J. Chem. Theory Comput.* **2009**, *5*, 2322−2338.
(33) Neelov, A.; Holm, C. *J. Chem. Phys.* **2010**, *132*, 234103.
(34) Ballenegger, V.; Cerdà, J. J.; Holm, C. *Comput. Phys. Commun.* **2011**, *182*, 1919−1923.
(35) Ballenegger, V.; Cerdà, J. J.; Holm, C. *J. Chem. Theory Comput.* **2012**, *8*, 936−947.
(36) Wang, H.; Schütte, C.; Zhang, P. *Phys. Rev. E* **2012**, *86*, 026704.
(37) Wang, H.; Zhang, P.; Schütte, C. *J. Chem. Theory Comput.* **2012**, *8*, 3243−3256.
(38) Ou-Yang, W. Z.; Lu, Z.-Y.; Shi, T. F.; Sun, Z. Y.; An, L. J. *J. Chem. Phys.* **2005**, *123*, 234502.
(39) Ismail, A. E.; Grest, G. S.; Stevens, M. J. *J. Chem. Phys.* **2006**, *125*, 014702.
(40) Ismail, A. E.; Tsige, M.; in 't Veld, P. J.; Grest, G. S. *Mol. Phys.* **2007**, *105*, 3155−3163.
(41) Wennberg, C. L.; Murola, T.; Hess, B.; Lindahl, E. *J. Chem. Theory Comput.* **2013**, *9*, 3527−3537.
(42) Jorgensen, W. J.; Maxwell, D. S.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1996**, *118*, 11225−11236.
(43) Martin, M. G.; Siepmann, J. I. *J. Phys. Chem. B* **1998**, *102*, 2569−2577.
(44) Scott, W. R. P.; Hünenberger, P. H.; Tironi, I. G.; Mark, A. E.; Billeter, S. R.; Fennen, J.; Torda, A. E.; Huber, T.; Krüger, P.; van Gunsteren, W. F. *J. Phys. Chem. A* **1999**, *103*, 3596−3607.
(45) Wolff, D.; Rudd, W. *Comput. Phys. Commun.* **1999**, *120*, 20−32.
(46) Tuckerman, M.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1992**, *97*, 1990−2001.
(47) Berendsen, H. J. C.; Grigera, J. R.; Straatsma, T. P. *J. Phys. Chem.* **1987**, *91*, 6269−6271.
(48) Kirkwood, J. G.; Buff, F. P. *J. Chem. Phys.* **1949**, *17*, 338−343.
(49) Plimpton, S. J. *Comput. Phys.* **1995**, *117*, 1−19.
(50) Berthelot, D. *C. R. Hebd. Acad. Sci.* **1898**, *126*, 1703−1855.
(51) Lorentz, H. A. *Ann. Phys.* **1881**, *12*, 127−136.
(52) Good, R. J.; Hope, C. J. *J. Chem. Phys.* **1970**, *53*, 540−543.
(53) Kong, C. L. *J. Chem. Phys.* **1973**, *59*, 2464−2467.
(54) Waldman, M.; Hagler, A. T. *J. Comput. Chem.* **1993**, *14*, 1077−1084.
(55) Skeel, R. D.; Tezcan, I.; Hardy, D. J. *J. Comput. Chem.* **2002**, *23*, 673−684.
(56) Tameling, D.; Springer, P.; Bientinesi, P.; Ismail, A. E. Multilevel summation for dispersion: A linear-time algorithm for $1/r^6$ potentials. arXiv preprint arXiv:1308.4005; 2013.
(57) Fumi, F. G.; Tosi, M. P. *J. Phys. Chem. Solids* **1964**, *25*, 31−44.
(58) Tosi, M. P.; Fumi, F. G. *J. Phys. Chem. Solids* **1964**, *25*, 45−52.
(59) Sorensen, R. A.; Liau, W. B.; Kesner, L.; Boyd, R. H. *Macromolecules* **1988**, *21*, 200−208.
(60) Borodin, O.; Smith, G. D.; Bedrov, D. *J. Phys. Chem. B* **2002**, *106*, 9912−9922.
(61) Zhang, L.; Siepmann, J. I. *J. Phys. Chem. B* **2005**, *109*, 2911−2919.
(62) Schnabel, T.; Srivastava, A.; Vrabec, J.; Hasse, H. *J. Phys. Chem. B* **2007**, *111*, 9871−9878.