

Reduced Computational Cost of Polarizable Force Fields by a Modification of the Always Stable Predictor-Corrector.

Dominique Nocito^{1, a)} and Gregory J. O. Beran¹

Department of Chemistry, University of California, Riverside, California 92521, United States

(Dated: 2 April 2019)

Classical polarizable force fields effectively incorporate the dynamic response of the electronic charge distributions into molecular dynamics simulations, but they do so at a significant increase in computational cost compared to simpler models. Here, we demonstrate how one can improve the stability of a polarizable force field molecular dynamics simulation or accelerate the evaluation of self-consistent polarization via a simple extension of the predictor in the Always Stable Predictor-Corrector (ASPC) method. Specifically, increasing the number of prior steps used in the predictor from six to sixteen reduces the energy drift by an order of magnitude. Alternatively, for a given level of energy drift, the induced dipoles can be obtained $\sim 20\%$ faster due to the reduced number of self-consistent field iterations required to maintain energetic stability. The extended-history predictor is straightforward to implement and involves minimal computational overhead.

Keywords: polarizable force fields, predictor, ASPC, AMOEBA, Tinker, Tinker-HP

Consideration of force field polarization is necessary to capture the transport properties of ionic liquids¹ and to describe protein structure adequately.^{2,3} However, the inclusion of polarization significantly increases the computational cost of classical molecular dynamics (MD) simulations. For instance, solving the large system of linear equations to obtain the induced dipoles in the AMOEBA force field⁴⁻⁶ accounts for about 50% of the computational cost of an MD simulation. In practice, this system of equations is too large to be solved exactly, and instead the solution is solved iteratively via a self-consistent field (SCF) method.⁷⁻⁹

In these SCF methods, successive iterations generally converge the induced dipoles toward their exact, mutually polarized values. The convergence thresholds for these SCF solvers must be chosen with care: When evaluating the polarization contributions to the nuclear forces, it is assumed that the iteratively determined induced dipoles have converged completely to the exact induced dipoles. Loose convergence of the induced dipoles can introduce instabilities in the simulation, such as problematic long-term energy conservation or deviations in physical properties.¹⁰ On the other hand, converging the induced dipoles more tightly via additional SCF iterations can increase the computational costs appreciably. Strategies based on perturbation theory,^{11,12} truncated conjugate gradients,^{13,14} and extended-Lagrangian models^{15,16} have been developed to circumvent the computational costs of converging the induced dipoles tightly during the polarization procedure.

Alternatively, use of a history-based predictor to construct a good initial guess for the SCF solver can significantly reduce the iterations and computational cost required to reach convergence.⁸ A predictor can provide an efficient means of calculating the induced dipoles with-

out introducing additional approximations. However, the use of induced dipoles from previous time steps destroys the time reversibility of the method.¹⁷ Time reversibility is a characteristic of symplectic integrators which preserve phase space volume and therefore conserve energy. A useful predictor should therefore exhibit an acceptable degree of time reversibility while substantially improving the starting point of the SCF method—i.e. it should provide an acceptable compromise between energy drift and computational cost. In this letter, we focus on the predictor from the Always Stable Predictor-Corrector (ASPC) method,¹⁸ which is a polynomial extrapolation scheme that employs information from previous time steps in a manner designed to guarantee stability during the MD time integration. Nevertheless, the ASPC predictor still introduces some energy drift in practice. Whereas conventional wisdom holds that the predictor should use information from only a relatively small number of recent steps, here we demonstrate how incorporating a much longer history in the predictor addresses stability concerns and/or reduces the computational cost of computing the induced dipoles by $\sim 20\%$.

The ASPC uses a history-based predictor for the induced dipoles,

$$\boldsymbol{\mu}^p(t+1) = \sum_{j=0}^{k+1} B_{j+1} \boldsymbol{\mu}(t-jh) \quad (1)$$

where $\boldsymbol{\mu}^p(t+1)$ is the predicted dipole, B_{j+1} are the scaling coefficients and $\boldsymbol{\mu}(t-jh)$ are the induced dipoles from previous time steps. The time step size is h and $k+2$ is the total number of values stored in history. The B_{j+1} scaling coefficients are derived such that the contributions that lead to time irreversibility error are chosen to be zero. In the original ASPC approach, the predicted induced dipoles are subsequently corrected by performing a single iteration of the SCF solver and then damping the resulting dipole update. The specific value

^{a)}Electronic mail: dominique.nocito@email.ucr.edu

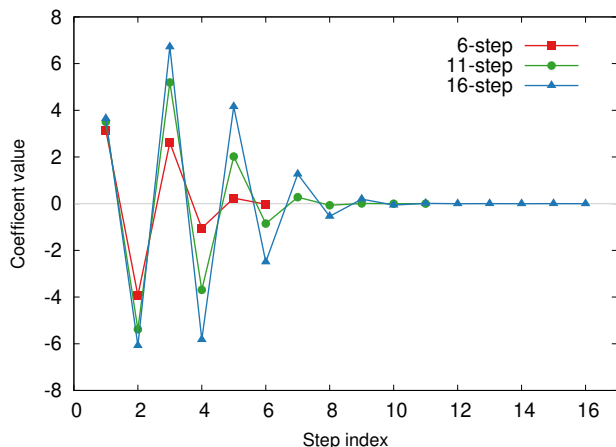


FIG. 1. Predictor coefficients with increasing history length.

of the damping coefficient is determined empirically, and its optimal value can potentially vary between systems and/or over the course of a simulation.

The Tinker software packages^{19–21} (and possibly others) avoid this empirical damping parameter part of the corrector. Instead they employ the so-called “predicted iteration” method,¹⁰ in which the predictor generates the initial guess for the induced dipoles, after which the SCF iterations are allowed to proceed until some user-defined convergence value is reached or a desired number of iterations has been performed. This predicted iteration variant of the ASPC is more accurate and circumvents the need to determine the optimal damping parameter.¹⁰

Previously, the predictor coefficients were worked out and tested up to the 6-step predictor ($k = 4$), but the 4-step predictor was suggested as a compromise between accuracy and memory storage.¹⁰ The additional SCF iterations performed in the predicted iteration method mitigate accumulation of error that might arise from the use of only a single SCF iteration in the ASPC. These additional SCF iterations potentially change the calculus regarding the optimal number of prior steps to include in the predictor, since a longer history might lead to a better guess for the dipoles and therefore require fewer iterations to converge at the next time step. Whereas the current implementation of this method in the Tinker packages employs a 6-step predictor, in this letter we test up to the 16-step predictor in the Tinker-HP v1.1 package.²⁰ The necessary coefficients for these higher N -step predictors can be derived from the recursive expressions presented previously.¹⁰ Using these expressions, we have derived the coefficients for up to a 25-step predictor, and these are included in the Supplementary Information.

Augmenting an existing implementation of the ASPC predictor to use higher N -step predictors is straightforward and adds little computational overhead. For a given system size, the memory requirements increase linearly with the number of induced dipole vectors stored in history. In a parallel implementation such as the one in

Tinker-HP,²⁰ these historical induced dipoles can be distributed across nodes, since each processor only needs knowledge of the dipole elements handled by that processor. Regardless, the total memory requirements are insignificant even with global storage: the induced dipole history for a 100,000 atom system for the 16-step predictor requires only 38.4 MB of memory in double precision. Evaluating the predictor requires just scalar multiplication, so the computational cost is negligible relative to the cost of an SCF iteration, and it scales linearly with the number of prior steps included.

In the ASPC predictor, the predicted induced dipole depends most strongly on the recent induced dipoles in history. In Figure 1 we see this trend holds for across a range of N -step predictors. In the 16-step predictor, for instance, the eight most recent history points account for 99.1% of the predicted induced dipole magnitude, while the oldest 8 history points contribute the remaining 0.9%. That means for a 1 fs time step, the 16-step predictor is dominated by contributions from the last 8 fs of simulation, which is shorter than the time period during which any substantial structural or conformational changes to the chemical system might occur.

The current work explores up to 16-step predictors, for which the B_{j+1} coefficients span seven orders of magnitude. The coefficients for the 25-step predictor span twelve orders of magnitude. The decision to stop at the 16-step predictor here is somewhat arbitrary. The 16-step predictor provides significant computational benefits (as shown below) while avoiding the need to handle many tiny contributions that would arise from employing a longer history.

We test the different N -step predictors here with the two SCF polarization solvers: the widely used preconditioned conjugate gradients (PCG) solver²² and our recently developed divide-and-conquer Jacobi iterations accelerated with direct inversion in the iterative subspace (DC-JI/DIIS) solver. We have previously demonstrated the superior speed and stability of DC-JI/DIIS relative to PCG.^{23,24} DC-JI/DIIS is used here unless otherwise specifically noted. Typically one iterates the SCF equations until a user-chosen convergence criterion is met. However, given the small numbers of iterations typically required to meet commonly used convergence criteria, even a change of one iteration arising from slightly different initial guesses can substantially alter how tightly converged the induced dipoles are. That in turn would obscure the stability behavior resulting from the predictor. To ensure an even-handed comparison of stability across the different solvers and predictors, all results here employ a fixed number of SCF iterations in the polarization solver. Testing indicates that the stability improvements reported here for the longer-history predictors also occur with more traditional threshold-based convergence criteria.

Stability of the predicted iteration approach is assessed here in terms of energy conservation in an NVE ensemble. The method also performs well for NVT ensembles,

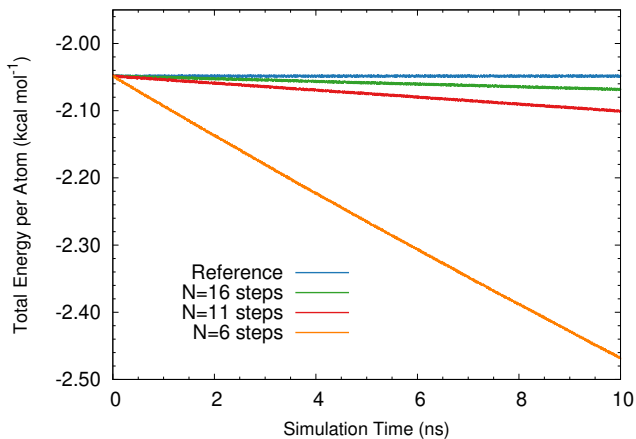


FIG. 2. Comparison of the energy conservation (in kcal mol^{-1} per atom) for NVE simulations on $(\text{H}_2\text{O})_{500}$ with different N -step predictors and three SCF iterations at each time step.

though using a thermostat obscures differences between the different predictors. Testing was done on a 500-molecule water box²⁵ and on the ubiquitin system.⁸ The 9,737-atom ubiquitin system consists of the 1,227-atom protein surrounded by 2,835 waters. All simulations were performed with the reversible reference system propagator algorithm (RESPA) multi-step integrator²⁶ using a 1 fs time step for non-bonded forces and 0.5 fs time step for the bonded forces. A direct space cutoff of 7 Å was employed for the particle-mesh Ewald treatment of long-range interactions. Energy drift was typically measured via linear regression of the energies over 1 ns of simulation time. For more tightly converged cases with less energy drift, 5–10 ns of simulation were used. Empirical testing indicates that these simulation lengths are sufficient to provide converged regression slopes (energy drift).

Figure 2 plots the energy conservation from NVE simulations on $(\text{H}_2\text{O})_{500}$ with different N -step predictors. The tightly converged reference simulation (i.e. 20 DC-JI/DIIS iterations, starting from initial guess dipoles equal to the polarizability times the permanent electric field) converges the dipoles to a root-mean-square change of $\sim 10^{-13}$ Debye, and it exhibits negligible drift ($< 10^{-5}$ $\text{kcal mol}^{-1} \text{ ns}^{-1} \text{ atom}^{-1}$). In contrast, the $N=6$ predictor with three SCF iterations per time step drifts by -0.042 $\text{kcal mol}^{-1} \text{ ns}^{-1} \text{ atom}^{-1}$. Increasing the length of the history employed in the predictor reduces the drift considerably. Despite taking only three SCF iterations per time step, the $N=16$ predictor case drifts by only -0.002 $\text{kcal mol}^{-1} \text{ ns}^{-1} \text{ atom}^{-1}$ over the 10 ns trajectory.

For a broader perspective, Figure 3 plots the drift per nanosecond in the $(\text{H}_2\text{O})_{500}$ box as a function of the number of steps included in the predictor and the number of SCF iterations. Each data point in this plot corresponds to a drift rate extracted from linear regression of an NVE simulation under those conditions. Independent of the

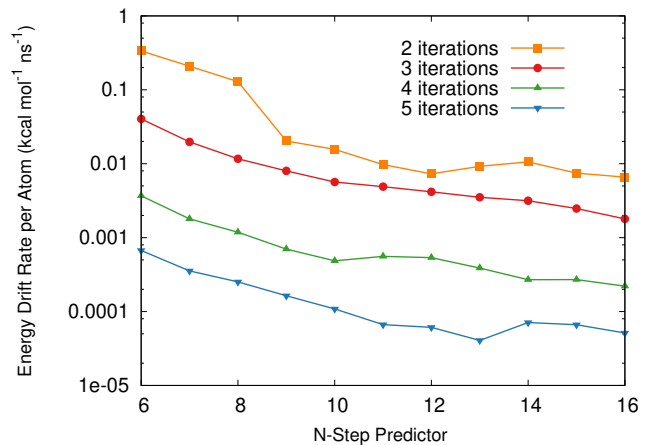


FIG. 3. Energy drift rate ($\text{kcal mol}^{-1} \text{ ns}^{-1}$) per atom for NVE simulations on $(\text{H}_2\text{O})_{500}$ for the 6-step to 16-step predictor with differing numbers of SCF iterations at each time step. A tightly converged reference simulation with no predictor exhibits energy drift less than 10^{-5} $\text{kcal mol}^{-1} \text{ ns}^{-1}$ per atom.

number of SCF iterations, increasing the history from the 6-step to the 16-step predictor decreases the energy drift rate by an order of magnitude. Moreover, the use of the 16-step predictor consistently improves the energy stability by an amount comparable to what one would obtain by performing an additional SCF iteration with the 6-step predictor. From that perspective, the better predictor can be used to accelerate the evaluation of the induced dipoles without increasing energy drift. For example, DC-JI/DIIS generally requires four SCF iterations to converge to a 10^{-5} Debye threshold.²⁴ With the 16-step predictor, comparable energy conservation can be obtained at the cost of only three SCF iterations, or a computational savings of $\sim 20\%$. The performance here is not unique to water, either. Similar energy drift behavior is observed for the ubiquitin system and an aqueous NaCl solution as well. See Supporting Information (SI).

It is interesting to compare the present approach with other recently developed strategies for accelerating polarizable force field simulations. For example, in the 16-step predicted iteration method with three iterations, the energy drift rate is only 0.002 $\text{kcal mol}^{-1} \text{ ns}^{-1} \text{ atom}^{-1}$. For comparison, a thermostatted extended-Lagrangian approach employing the same number of SCF iterations exhibited a somewhat larger energy drift of ~ 0.009 $\text{kcal mol}^{-1} \text{ ns}^{-1} \text{ atom}^{-1}$ for a similar water box.¹⁵ The approximate OPT3 perturbative polarization solver¹² also effectively utilizes three SCF iterations, but it requires several empirically fitted parameters to achieve good accuracy. Furthermore, the large N -step predictor approach here is probably as fast or faster than the truncated conjugate gradient approximate solvers (at least for 1 fs time steps), since those effectively employ 2–3 PCG iterations and have more expensive analytic gradients.¹⁴

On the other hand, the performance of the traditional ASPC predictor implementation degrades when longer

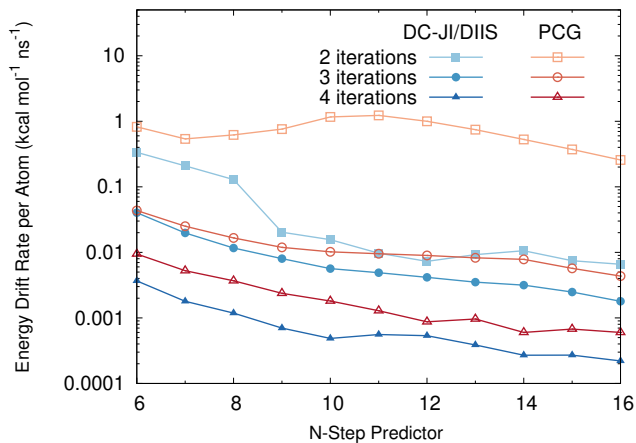


FIG. 4. Comparison of different higher N -step predictors with the DC-JI/DIIS and PCG polarization solvers.

time steps are used in the simulation, unlike solvers such as OPT or TCG where the number of SCF steps depends only on the current state and is independent of time step size. Results are presented in the SI examining the behavior of longer-history predictor with 3 fs time steps instead of 1 fs ones (using hydrogen mass reweighting as implemented in Tinker-HP). With 3 fs time steps, two additional DC-JI/DIIS SCF iterations are required to maintain energy drift comparable to that of 1 fs time steps. Increasing the history from 6 to 16 steps still improves energy conservation noticeably, though the magnitude is reduced from ~ 10 -fold with a 1 fs step to ~ 3 –4-fold with a 3 fs time step. In other words, the benefits of the longer-history predictor decrease when longer time steps are taken, but they are still appreciable and worthwhile given the low computational cost associated with them.

A direct and thorough performance comparison among the different polarization approaches over a range of simulation scenarios would be a valuable future work. Here, we compare just two polarization solvers, DC-JI/DIIS and PCG, with higher N -step predictors in Figure 4. With 1 fs time steps, both SCF methods generally benefit from employing the higher N -step predictors. However, DC-JI/DIIS exhibits less drift relative to PCG for all simulations. Interestingly, for the case of the PCG solver using two iterations, using higher N -step predictors does not decrease the energy drift, in contrast to what is observed for the more robust DC-JI/DIIS solver. For the 3 fs time step case examined in SI, PCG exhibits no benefit from the longer predictor. Both observations contrast the behavior of the more robust DC-JI/DIIS solver, where the longer predictor is always beneficial. PCG is more sensitive to the initial guess dipoles than other methods like JI.⁸

Finally, to understand why the longer N -step predictors perform better than shorter-history ones, Figure 5 shows the distribution of dipole errors in the ini-

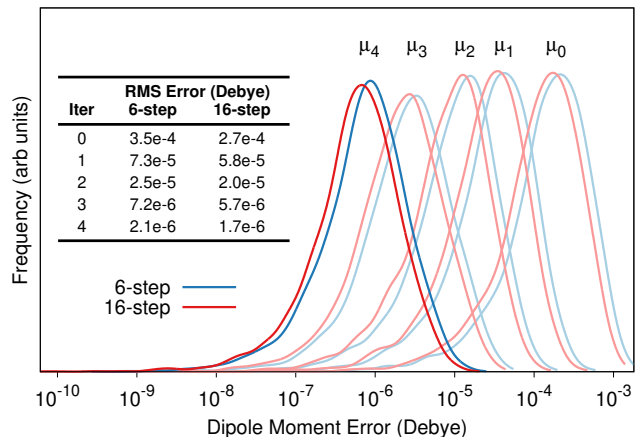


FIG. 5. Distributions of errors in the induced dipoles from the initial guess (μ_0) and first four SCF iterations relative to tightly converged dipoles using either the 6-step (blue) or 16-step predictor (red).

tial guess (μ_0) and after each successive iteration relative to a tightly converged (20 SCF iterations) reference set. With both 6-step and 16-step predictors, the initial guess dipoles have errors around 10^{-4} D, but the root-mean-square (rms) errors for the 16-step case are about $\sim 20\%$ smaller. The errors in the induced dipoles decrease several fold with each SCF iteration, but the dipoles from the 16-step predictor consistently maintain $\sim 20\%$ higher accuracy. These small accuracy improvements in the induced dipoles are sufficient to increase the stability of the simulations significantly.

In conclusion, we have demonstrated that use of a longer history in the “predicted iteration” variant of the ASPC provides substantial computational benefits in the context of the AMOEBA force field. Increasing the history from 6 to 16 steps requires only minor software modifications and adds little computational overhead, yet it reduces the rate of energy drift by an order of magnitude. Alternatively, one can employ this strategy to reduce the number of SCF iterations and accelerate the calculation of force field polarization by $\sim 20\%$. The ability to achieve acceptable energy conservation with only three SCF iterations makes the combination of the 16-step predictor and DC-JI/DIIS polarization solver competitive with other approximate and extended-Lagrangian schemes for handling the induced dipoles. The extended predictor should prove useful for other polarizable force fields in addition to AMOEBA, and perhaps it could also be adapted for *ab initio* molecular dynamics simulations that employ the ASPC.²⁷ In addition, preliminary testing with higher-order Gear predictors did not provide the same reduction in energy drift seen here for the longer-history ASPC predictor. In the future, it might be interesting to investigate behaviors of different predictors in this context in more detail.

ACKNOWLEDGMENTS

Funding for this work from the National Science Foundation (CHE-1665212) and supercomputer time from XSEDE (TG-CHE110064) are gratefully acknowledged. D.N. was supported by a Software Fellowship from the Molecular Sciences Software Institute^{28,29} under NSF grant (ACI-1547580).

- ¹O. Borodin, J. Phys. Chem. B **113**, 11463 (2009).
- ²J. Huang and A. D. MacKerell, Curr. Opin. Struct. Bio. **48**, 40 (2018).
- ³D. S. Davidson, A. M. Brown, and J. A. Lemkul, J. Mol. Biol. **430**, 3819 (2018).
- ⁴P. Ren and J. W. Ponder, J. Phys. Chem. B **107**, 5933 (2003).
- ⁵P. Ren, C. Wu, and J. W. Ponder, J. Chem. Theory Comput. **7**, 3143 (2011).
- ⁶Y. Shi, Z. Xia, J. Zhang, R. Best, C. Wu, J. W. Ponder, and P. Ren, J. Chem. Theory Comput. **9**, 4046 (2013).
- ⁷W. Wang and R. D. Skeel, J. Chem. Phys. **123**, 164107 (2005).
- ⁸F. Lipparini, L. Lagardère, B. Stamm, E. Cancès, M. Schnieders, P. Ren, Y. Maday, and J.-P. Piquemal, J. Chem. Theory Comput. **10**, 1638 (2014).
- ⁹L. Lagardère, F. Lipparini, E. Polack, B. Stamm, E. Cancès, M. Schnieders, P. Ren, Y. Maday, and J.-P. Piquemal, J. Chem. Theory Comput. **11**, 2589 (2015).
- ¹⁰J. Kolafa, J. Comp. Chem. **25**, 335 (2003).
- ¹¹A. C. Simmonett, F. C. Pickard IV, Y. Shao, T. E. Cheatham III, and B. R. Brooks, J. Chem. Phys. **143**, 074115 (2015).
- ¹²A. C. Simmonett, F. C. Pickard, J. W. Ponder, and B. R. Brooks, J. Chem. Phys. **145**, 164101 (2016).
- ¹³F. Aviat, A. Levitt, B. Stamm, Y. Maday, P. Ren, J. W. Ponder, L. Lagardère, and J.-P. Piquemal, J. Chem. Theory Comput. **13**, 180 (2017).
- ¹⁴F. Aviat, L. Lagardère, and J.-P. Piquemal, J. Chem. Phys. **147**, 161724 (2017).
- ¹⁵A. Albaugh, O. Demerdash, and T. Head-Gordon, J. Chem. Phys. **143**, 174104 (2015).
- ¹⁶A. Albaugh, A. M. N. Niklasson, and T. Head-Gordon, J. Phys. Chem. Lett. **8**, 1714 (2017).
- ¹⁷D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, San Diego, 2002).
- ¹⁸J. Kolafa, Mol. Sim. **18**, 193 (1996).
- ¹⁹J. A. Rackers, Z. Wang, C. Lu, M. L. Laury, L. Lagardère, M. J. Schnieders, J.-P. Piquemal, P. Ren, and J. W. Ponder, J. Chem. Theory Comput. **14**, 5273 (2018).
- ²⁰L. Lagardère, L.-H. Jolly, F. Lipparini, F. Aviat, B. Stamm, Z. F. Jing, M. Harger, H. Torabifard, G. A. Cisneros, M. J. Schnieders, N. Gresh, Y. Maday, P. Y. Ren, J. W. Ponder, and J.-P. Piquemal, Chem. Sci. **9**, 956 (2018).
- ²¹M. Harger, D. Li, Z. Wang, K. Dalby, L. Lagardère, J.-P. Piquemal, J. Ponder, and P. Ren, J. Comp. Chem. **38**, 2047 (2017).
- ²²W. Wang and R. D. Skeel, J. Chem. Phys. **123**, 164107 (2005).
- ²³D. Nocito and G. J. O. Beran, J. Chem. Phys. **146**, 114103 (2017).
- ²⁴D. Nocito and G. J. O. Beran, J. Chem. Theory Comput. **14**, 3633 (2018).
- ²⁵Tinker example directory, <https://dasher.wustl.edu/tinker/distribution/example/>, accessed: 2018-11-10.
- ²⁶M. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **97**, 1990 (1992).
- ²⁷B. Kirchner, P. J. di Dio, and J. Hutter, in *Multiscale Molecular Methods in Applied Chemistry*, edited by B. Kirchner and J. Vrabec (Springer, Berlin, Heidelberg, 2012), pp. 109–153.
- ²⁸N. Wilkins-Diehr and T. D. Crawford, Comput. Sci. Eng. **20**, 26 (2018).
- ²⁹A. Krylov, T. L. Windus, T. Barnes, E. Marin-Rimoldi, J. A. Nash, B. Pritchard, D. G. A. Smith, D. Altarawy, P. Saxe, C. Clementi, T. D. Crawford, R. J. Harrison, S. Jha, V. S. Pande, and T. Head-Gordon, J. Chem. Phys. **149**, 180901 (2018).
- ³⁰O. Borodin, J. Phys. Chem. B **113**, 11463 (2009).
- ³¹J. Huang and A. D. MacKerell, Curr. Opin. Struct. Bio. **48**, 40 (2018).
- ³²D. S. Davidson, A. M. Brown, and J. A. Lemkul, J. Mol. Biol. **430**, 3819 (2018).
- ³³P. Ren and J. W. Ponder, J. Phys. Chem. B **107**, 5933 (2003).
- ³⁴P. Ren, C. Wu, and J. W. Ponder, J. Chem. Theory Comput. **7**, 3143 (2011).
- ³⁵Y. Shi, Z. Xia, J. Zhang, R. Best, C. Wu, J. W. Ponder, and P. Ren, J. Chem. Theory Comput. **9**, 4046 (2013).
- ³⁶W. Wang and R. D. Skeel, J. Chem. Phys. **123**, 164107 (2005).
- ³⁷F. Lipparini, L. Lagardère, B. Stamm, E. Cancès, M. Schnieders, P. Ren, Y. Maday, and J.-P. Piquemal, J. Chem. Theory Comput. **10**, 1638 (2014).
- ³⁸L. Lagardère, F. Lipparini, E. Polack, B. Stamm, E. Cancès, M. Schnieders, P. Ren, Y. Maday, and J.-P. Piquemal, J. Chem. Theory Comput. **11**, 2589 (2015).
- ³⁹J. Kolafa, J. Comp. Chem. **25**, 335 (2003).
- ⁴⁰A. C. Simmonett, F. C. Pickard IV, Y. Shao, T. E. Cheatham III, and B. R. Brooks, J. Chem. Phys. **143**, 074115 (2015).
- ⁴¹A. C. Simmonett, F. C. Pickard, J. W. Ponder, and B. R. Brooks, J. Chem. Phys. **145**, 164101 (2016).
- ⁴²F. Aviat, A. Levitt, B. Stamm, Y. Maday, P. Ren, J. W. Ponder, L. Lagardère, and J.-P. Piquemal, J. Chem. Theory Comput. **13**, 180 (2017).
- ⁴³F. Aviat, L. Lagardère, and J.-P. Piquemal, J. Chem. Phys. **147**, 161724 (2017).
- ⁴⁴A. Albaugh, O. Demerdash, and T. Head-Gordon, J. Chem. Phys. **143**, 174104 (2015).
- ⁴⁵A. Albaugh, A. M. N. Niklasson, and T. Head-Gordon, J. Phys. Chem. Lett. **8**, 1714 (2017).
- ⁴⁶D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, San Diego, 2002).
- ⁴⁷J. Kolafa, Mol. Sim. **18**, 193 (1996).
- ⁴⁸J. A. Rackers, Z. Wang, C. Lu, M. L. Laury, L. Lagardère, M. J. Schnieders, J.-P. Piquemal, P. Ren, and J. W. Ponder, J. Chem. Theory Comput. **14**, 5273 (2018).
- ⁴⁹L. Lagardère, L.-H. Jolly, F. Lipparini, F. Aviat, B. Stamm, Z. F. Jing, M. Harger, H. Torabifard, G. A. Cisneros, M. J. Schnieders, N. Gresh, Y. Maday, P. Y. Ren, J. W. Ponder, and J.-P. Piquemal, Chem. Sci. **9**, 956 (2018).
- ⁵⁰M. Harger, D. Li, Z. Wang, K. Dalby, L. Lagardère, J.-P. Piquemal, J. Ponder, and P. Ren, J. Comp. Chem. **38**, 2047 (2017).
- ⁵¹W. Wang and R. D. Skeel, J. Chem. Phys. **123**, 164107 (2005).
- ⁵²D. Nocito and G. J. O. Beran, J. Chem. Phys. **146**, 114103 (2017).
- ⁵³D. Nocito and G. J. O. Beran, J. Chem. Theory Comput. **14**, 3633 (2018).
- ⁵⁴Tinker example directory, <https://dasher.wustl.edu/tinker/distribution/example/>, accessed: 2018-11-10.
- ⁵⁵M. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **97**, 1990 (1992).
- ⁵⁶B. Kirchner, P. J. di Dio, and J. Hutter, in *Multiscale Molecular Methods in Applied Chemistry*, edited by B. Kirchner and J. Vrabec (Springer, Berlin, Heidelberg, 2012), pp. 109–153.
- ⁵⁷N. Wilkins-Diehr and T. D. Crawford, Comput. Sci. Eng. **20**, 26 (2018).
- ⁵⁸A. Krylov, T. L. Windus, T. Barnes, E. Marin-Rimoldi, J. A. Nash, B. Pritchard, D. G. A. Smith, D. Altarawy, P. Saxe, C. Clementi, T. D. Crawford, R. J. Harrison, S. Jha, V. S. Pande, and T. Head-Gordon, J. Chem. Phys. **149**, 180901 (2018).