Adaptively Restrained Particle Simulations

Svetlana Artemova and Stephane Redon

NANO-D, INRIA Grenoble-Rhone-Alpes, 38334 Saint Ismier Cedex, Montbonnot, France and Laboratoire Jean Kuntzmann, B.P. 53, 38041 Grenoble Cedex 9, France

(Received 4 February 2012; revised manuscript received 22 June 2012; published 7 November 2012)

Interaction potentials used in particle simulations are typically written as a sum of terms which depend on just a few relative particle positions. Traditional simulation methods move all particles at each time step, and may thus spend a lot of time updating interparticle forces. In this Letter we introduce adaptively restrained particle simulations (ARPS) to speed up particle simulations by adaptively switching on and off positional degrees of freedom, while letting momenta evolve. We illustrate ARPS on several numerical experiments, including (a) a collision cascade example that demonstrates how ARPS make it possible to smoothly trade between precision and speed and (b) a polymer-in-solvent study that shows how one may efficiently determine static equilibrium properties with ARPS.

DOI: 10.1103/PhysRevLett.109.190201 PACS numbers: 02.70.Ns, 05.10.−a, 07.05.Tp, 45.20.Jj

Particle simulations are widely used in physics, chemistry, biology [1,2], and even computer graphics [3]. However, many important problems still constitute significant computational challenges, including molecular docking, protein folding, diffusion across biomembranes, fracture in metals, ion implantation, etc. Numerous methods have been developed to accelerate particle simulations by, e.g., increasing the simulation's time step [4–9], improving the computational complexity of the simulation [10–14], or simplifying the system under study [14–19], in particular, via coarse-graining methods [20–22] or multiscale and multiresolution methods [23–26].

In this Letter, we introduce a novel, general approach to speed up particle simulations that we call adaptively restrained particle simulations (ARPS). Our approach adaptively switches positional degrees of freedom on and off during a simulation, while letting the corresponding momenta evolve. The benefits of this approach are that (a) it is mathematically grounded and is able to produce long, stable simulations, (b) it does not requires modifications to the simulated interaction potential, so that any suitable existing force field can be directly used with ARPS, (c) under frequently used assumptions on the interaction potential, ARPS make it possible to reduce the number of forces that have to be updated at each time step, which may significantly speed up simulations, (d) when performing constant-energy simulations, ARPS allow users to finely and continuously trade between precision and computational cost, and rapidly obtain approximate trajectories, (e) the trade-off between precision and cost may be chosen for each particle independently, so that users may arbitrarily focus ARPS on specific regions of the simulated system (e.g., a polymer in a solvent), (f) most importantly, when performing adaptively restrained molecular dynamics (ARMD) in the canonical (NVT) ensemble, correct static equilibrium properties can be computed.

The dynamics of a system of N particles in 3D is commonly derived from a Hamiltonian function [27]:

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2}\,\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p} + V(\mathbf{q}), \tag{1}$$

where $(\mathbf{q}, \mathbf{p})$ is a vector of $6N$ phase space coordinates, $\mathbf{M}$ is a $3N \times 3N$ mass matrix (which might depend on $\mathbf{q}$), the first term of the sum represents the kinetic energy of the system, and $V(\mathbf{q})$ the interaction potential. This potential (derived from classical or quantum mechanical computations) is typically represented as a sum of terms that only depend on a few relative positional degrees of freedom in the system [28–30].

In our approach, we introduce an adaptively restrained (AR) Hamiltonian, i.e., a Hamiltonian with a modified inverse inertia matrix $\Phi(\mathbf{q}, \mathbf{p})$:

$$H_{\mathrm{AR}}(\mathbf{q}, \mathbf{p}) = \frac{1}{2}\,\mathbf{p}^T\Phi(\mathbf{q}, \mathbf{p})\mathbf{p} + V(\mathbf{q}), \tag{2}$$

and we use the matrix $\Phi(\mathbf{q}, \mathbf{p})$ to specify how and when positional degrees of freedom are switched on and off during the simulation. In this Letter, we propose such a matrix for particle simulations in Cartesian coordinates.

How to switch.—We choose to switch positional degrees of freedom on and off for each particle independently, and, to do so, we use a diagonal matrix $\Phi(\mathbf{q}, \mathbf{p})$. Each diagonal 3×3 block of this matrix corresponds to a single particle i, and is an identity matrix multiplied by the particle's adaptive inverse inertia $\phi_i(q_i, p_i)$. We choose $\phi_i(q_i, p_i) = m_i^{-1}[1 - \rho_i(q_i, p_i)]$, $1 \le i \le N$, where m_i is the particle's mass, and $\rho_i(q_i, p_i) \in [0, 1]$ is a twice-differentiable restraining function. When $\rho_i(q_i, p_i) = 0$ (no restraining), $\phi_i(q_i, p_i) = m_i^{-1}$ and the particle follows full dynamics [the dynamics derived from the Hamiltonian Eq. (1)]. When $\rho_i(q_i, p_i) = 1$ (full restraining), $\phi_i(q_i, p_i) = 0$ and the particle is not moving, whatever the force applied to it.

When $\rho_i(q_i, p_i) \in (0, 1)$, the particle smoothly switches between both behaviors.

When to switch.—We make the restraining function of each particle i depend on its kinetic energy $K_i = p_i^2/(2m_i)$, so that $\rho_i(q_i, p_i) = \rho_i(p_i)$. Let ε_i^r and ε_i^f, $\varepsilon_i^r < \varepsilon_i^f$, respectively, denote a restrained-dynamics threshold and a full-dynamics threshold for particle i. We define $\rho_i(p_i)$ as:

$$\rho_i(p_i) = \begin{cases} 1, & \text{if } 0 \le K_i(p_i) \le \varepsilon_i^r, \\ 0, & \text{if } K_i(p_i) \ge \varepsilon_i^f, \\ s(K_i(p_i)) \in [0, 1], & \text{elsewhere,} \end{cases}$$

where $s(K_i)$ is a twice-differentiable function with respect to its argument and, therefore, to p_i.

The equations of motion can now be derived from the AR Hamiltonian:

$$\dot{\mathbf{p}} = -\frac{\partial H_{\mathrm{AR}}}{\partial \mathbf{q}} = -\frac{\partial V(\mathbf{q})}{\partial \mathbf{q}},$$
$$\dot{\mathbf{q}} = \frac{\partial H_{\mathrm{AR}}}{\partial \mathbf{p}} = \mathbf{M}^{-1}[\mathbf{I} - \rho(\mathbf{p})]\mathbf{p} - \frac{1}{2}\,\mathbf{p}^T\mathbf{M}^{-1}\frac{\partial \rho(\mathbf{p})}{\partial \mathbf{p}}\,\mathbf{p}, \tag{3}$$

where $\mathbf{I}$ is an identity matrix, $\rho(\mathbf{p})$ is a $3N \times 3N$ diagonal matrix combining the individual $\rho_i(p_i)$. As with Hamiltonian [Eq. (1)], the time derivative of the momenta vector is equal to the negative gradient of the potential. Positions, however, evolve differently. When a particle's momentum becomes small enough (without necessarily becoming zero), the particle completely stops moving. Even when a particle is fully-restrained, though, its momentum may continue to change, and its kinetic energy might become large enough again for the particle to resume moving. In general, ARPS restrain and release particles repeatedly over time.

Consider for example a 1D harmonic oscillator, i.e., a single particle of mass 1 g/mol attached with a spring of stiffness 1 kcal/(mol Å^2) to the origin. Figure 1 shows a phase space portrait of the corresponding AR system. Gray lines stand for isolines of the AR Hamiltonian, the thick black line—for a specific isoline of this Hamiltonian ($H_{\mathrm{AR}} \equiv 1$), the dotted red circle represents the corresponding isovalue of the classical Hamiltonian ($H \equiv 1$). The restrained-dynamics region (where the particle is fully-restrained) is blue, the full-dynamics region (where the particle is not restrained) is green, and black dashed lines indicate the boundaries of these two regions. The AR Hamiltonian has modified trajectories: in the restrained-dynamics region positions are constant, but momenta change; in the transition region trajectories smoothly switch between restrained dynamics and full dynamics. When $p = 0$, the AR trajectory is tangential to the classical one.

In this Letter, we validate our method by considering molecular dynamics in two widely used thermodynamical ensembles. Hamiltonian dynamics samples the microcanonical ensemble (NVE). However, statistical averages are often computed in the canonical (NVT) ensemble. In this ensemble, ARPS can determine any static equilibrium property $\langle \mathbf{A} \rangle_H$ by considering the AR Hamiltonian as a biased version of the original one: $H_{\mathrm{AR}} = H + V_{\mathrm{AR}}(\mathbf{q}, \mathbf{p})$. Phase space samples generated according to the AR distribution are then weighted [31]:

$$\langle \mathbf{A} \rangle_H = \langle \mathbf{A} e^{(V_{\mathrm{AR}}/k_B T)} \rangle_{H_{\mathrm{AR}}} / \langle e^{(V_{\mathrm{AR}}/k_B T)} \rangle_{H_{\mathrm{AR}}}.$$

When $\mathbf{A}$ only depends on positions, and the AR Hamiltonian is separable (as in this Letter), the correct statistics are even obtained straight away: $\langle \mathbf{A} \rangle_{H_{\mathrm{AR}}} = \langle \mathbf{A} \rangle_H$.

Computational performance.—AR simulations may result in significant speed-ups when the particle forces can be incrementally updated at each time step based on the list of particles that have moved. Such incremental update algorithms have been developed before [30,32]. In this Letter, we propose a simple method to efficiently update forces under two frequent assumptions: (a) the interaction potential is a sum of pairwise terms (thus, there is no need to update forces between two fully-restrained particles) and (b) a cutoff distance is used. The method relies on a 3D grid of cells, to which all particles are distributed according to their positions [27]. Thanks to the grid, all particles that interact with a specific particle in cell c can be found in a limited number of cells around cell c. Every time step, for the whole set of active (non fully-restrained) particles, we incrementally update applied forces: (1) all

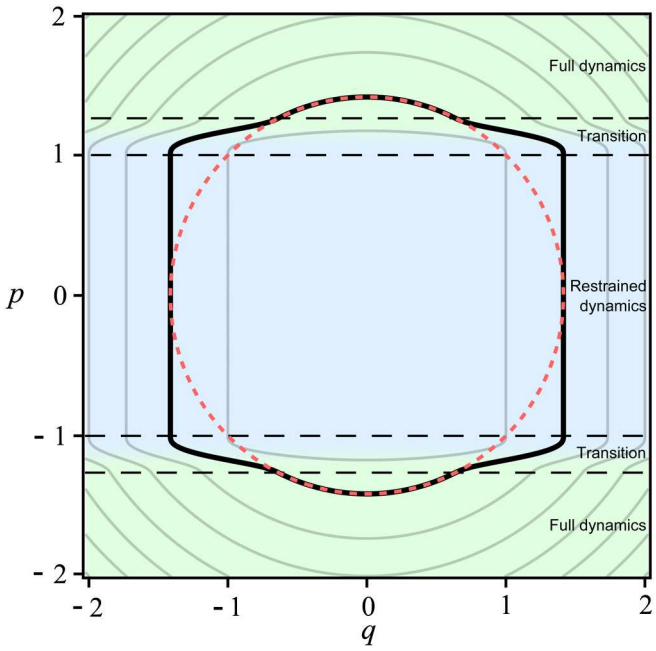

FIG. 1 (color online). Phase portrait of the adaptively restrained harmonic oscillator ($\varepsilon_1^r = 0.5$, $\varepsilon_1^f = 0.8$ kcal/mol). In the full-dynamics region, trajectories remain unperturbed. In the restrained-dynamics region, positions are constant although momenta evolve.

forces that were acting on each active particle at the previous time step (based on previous positions) are subtracted, (2) the grid is updated, since some active particles may have moved to other cells, (3) new forces, based on current positions, are added to each active particle. This method has a linear time complexity in the number of active particles.

Implementation and results.—We designed four numerical tests to validate our approach and demonstrate the benefits of ARPS in the context of molecular dynamics. For *NVE* simulations, we used a partitioned Euler method [33] to integrate the equations of motion [Eq. (3)]. Since, in this Letter, the AR Hamiltonian is separable, this integrator is symplectic and explicit. For *NVT* simulations, we used a Langevin thermostat in the general form [31], and the following integration splitting scheme: a half step for the Langevin part of the equations, a full step for the Hamiltonian part, and again, a half step for the Langevin part [34]. This integrator is implicit. To solve the nonlinear equation for momenta, we used the fixed-point method: in practice, about three iterations were sufficient for this method to converge.

The ARPS method was implemented in C++ and tested either on Computer 1 (two Intel Xeon X5450 3 GHz quad-core processors, 16 GB of RAM, Windows Vista 64-bit operating system) or on Computer 2 (one Intel 2.40 GHz quad-core processor, 4 GB of RAM, Windows Vista 32-bit OS). The implementation was serial: each simulation only used one processing core at a time.

Argon liquid.—To show that ARPS is able to produce long stable trajectories, we simulated a periodic 3D box of 21952 Argon particles (box length 99.3048 Å, particle mass 39.95 g/mol) in the *NVE* ensemble. A Lennard-Jones potential [35] was used ($\epsilon/k_B = 120$ K, $\sigma = 3.4$ Å, cutoff = 8 Å, the potential was truncated through a smoothing function applied between 7.5 and 8 Å, the same parameters used for all presented tests). We performed three simulations: a reference (full-dynamics) simulation, and two AR simulations with different thresholds (time step size 0.488 fs, 5000000 time steps, total simulation time 2.44 nanoseconds, Computer 1). Figure 2 shows the system's total adaptive energy for both AR simulations, i.e., the value of H_{AR} over time. Thanks to the readily available symplectic integrator, these adaptive energies are stable. Note how the total adaptive energies do not coincide with each other, nor with the total energy of the reference simulation ($-23\,459$ kcal/mol, not plotted): even though the same initial conditions were used for both positions and momenta, different Hamiltonians were simulated and preserved. For both AR simulations, Fig. 2 reports (a) the average number of active particles $\langle N_{\mathrm{act}} \rangle$ as a percentage of the total number, (b) the average number of updated forces $\langle N_F \rangle$ relatively to the reference simulation (which required about 537 000 force updates on average, per time step), and (c) the total speed-up with respect to the reference

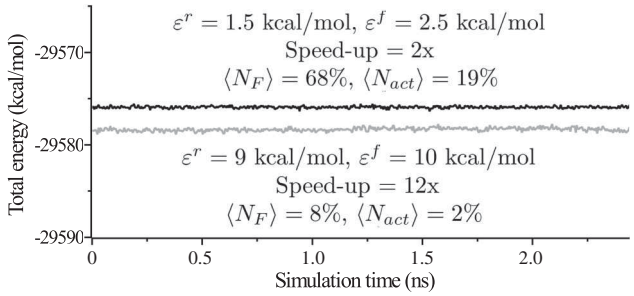

FIG. 2. Adaptively restrained simulations of a periodic 3D box of 21 952 Argon particles, with two different pairs of thresholds (*NVE* ensemble). The total adaptive energies are stable for both simulations, even when the average number of active particles $\langle N_{\mathrm{act}} \rangle$ is greatly reduced. The decrease in the average number of updated forces $\langle N_F \rangle$ results in significant speed-ups.

simulation (which took about 10 CPU days), due to the reduction in $\langle N_F \rangle$.

Collision cascade.—To demonstrate how AR particle simulations allow us to smoothly trade between computational cost and precision, we simulated a collision cascade in a 2D system composed of 5930 particles with mass 1 g/mol in the *NVE* ensemble, using again a Lennard-Jones potential. We performed four simulations of a shock induced by a particle launched at high velocity towards an initially static 2D system: a reference (full-dynamics) simulation, and three AR simulations with varying degrees of precision (time step size 0.0488 fs, 7000 steps, total simulation time 342 fs, Computer 2). Figure 3 compares the final configurations reached by the simulations. For each AR simulation, the root-mean-square deviation (RMSD) from the reference final configuration is given, as well as the maximum particle displacement error Δq_{max}. In this example, AR simulations allow for large speed-ups (up to 10 times) while preserving the features of the shock extremely well.

Radial distribution function.—To show that ARPS may be used to compute static equilibrium properties in the *NVT* ensemble, we performed Langevin simulations of a system of 343 Argon particles interacting in a 3D periodic box through a Lennard-Jones potential (box length 25.56 Å, Langevin friction coefficient $\gamma = 1$, $T = 94.4$ K). The particles were initially far from equilibrium, positioned at the nodes of a 3D cubical lattice. We launched a full-dynamics simulation and an AR simulation and computed the radial distribution function $g(r)$ in each case (time step size 0.488 fs, 150 000 steps, total simulation length 73.2 ps, Computer 2). Figure 4 plots the results: even though the AR simulation has significantly modified the system's dynamics (only 3% of the particles are moving on average at each time step), the curves coincide, demonstrating that the equilibrium statistics have been preserved.

Polymer in solvent.—Finally, to show how ARPS may be used to obtain static properties faster than with full-dynamics simulations, we performed Langevin simulations

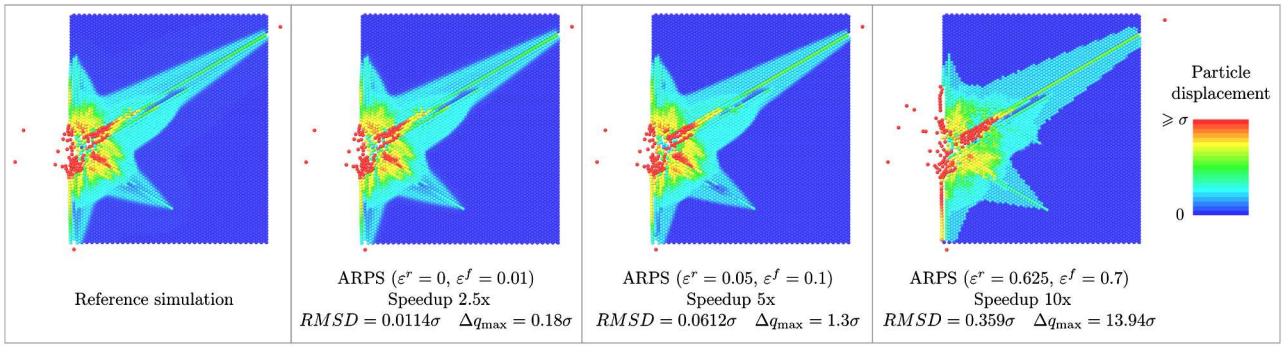

FIG. 3 (color online). Simulating a collision cascade with controlled precision (thresholds in kcal/mol, NVE ensemble). AR simulations allow us to smoothly trade between precision and speed. Even for large speed-ups (up to 10 times), the features of the picture are extremely well preserved.

of a toy polymer in a small solvent box (343 Argon particles, box length 25.56 Å) in order to predict its hydrodynamic radius ($\gamma = 1$, $T = 350$ K). This numerical experiment is representative of numerous applications of particle simulations in physics, chemistry, biology, etc., where information is collected about a small part of the particle system, but where the rest of the system is required so that the collected information is realistic (e.g., simulating an active site in an enzyme, a solute passing through a membrane channel, a defect in a graphene sheet, a crack in a material, etc.). We modeled the polymer as five particles of mass 40 g/mol each, connected by springs (stiffness $k = 30\,000\ \epsilon/\sigma^2$, equilibrium length $d_0 = 0.07071 \times \sigma = 2.4$ Å). Here σ is the parameter of the Lennard-Jones potential described above in the Argon liquid example. The solvent interacted with the polymer through a Lennard-Jones potential (same set of parameters).

Since we wanted to obtain statistics on the polymer, we only restrained the solvent particles ($\varepsilon^r = 18$ kcal/mol, $\varepsilon^f = 20$ kcal/mol). We compared the rate of convergence of the polymer's hydrodynamic radius R_H to its average value $\langle R_H \rangle$ in full-dynamics simulations and AR simulations. To achieve this, despite the intrinsic stochasticity of Langevin simulations, we performed 90 day-long full-dynamics simulations and 90 day-long AR simulations (time step size 0.488 fs, a total of 6 CPU months, Computer 1). The values of $\langle R_H \rangle$ averaged over 90 simulations are very close (4.459, and 4.462 Å, respectively).

Figure 5 plots the variance of R_H in full-dynamics simulations and AR simulations, first as a function of simulation time (top) and then as a function of wall-clock time (bottom). In simulation time (i.e., per time step), the full-dynamics variance decreased faster than the AR variance. This is understandable: solvent particles, which influence the polymer conformations, were often static in the AR simulations (2% of active particles on average), so that phase space was less efficiently sampled at each time step. The computational cost of AR time steps, however,

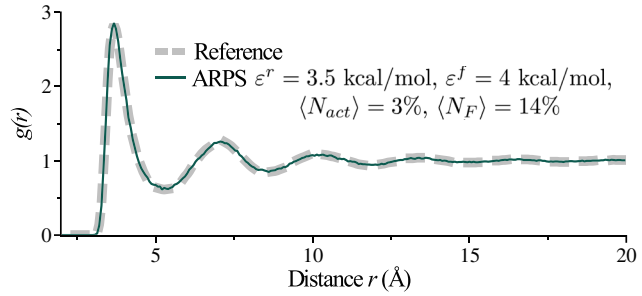

FIG. 4 (color online). Radial distribution functions for an Argon system (periodic 3D box, 343 particles, NVT ensemble). Even though the AR simulation has significantly modified the system's dynamics (see $\langle N_{\text{act}} \rangle$), the curves coincide.

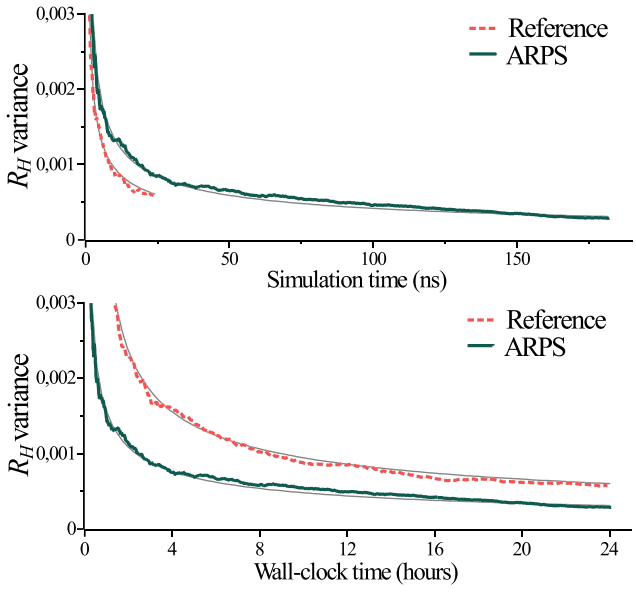

FIG. 5 (color online). Computing the hydrodynamic radius of a solvated polymer (NVT ensemble). Full-dynamics simulations reduce the variance more at each time step (top), but AR simulations perform many more time steps, so that they reduce the variance faster in wall-clock time (bottom). For any target precision, AR simulations compute the hydrodynamic radius four times faster than full-dynamics simulations.

was significantly reduced: AR simulations allowed for about seven times more time steps in the same wall-clock duration (3 645 000 vs 495 000). As a result, the AR variance decreased faster than the full-dynamics variance per wall-clock second. To estimate the speed-up, we fitted the curves with functions of the form: $a/\sqrt{x+b}$. We obtained $a = 0.0029$ ($b = -0.484$) for the reference simulations and $a = 0.0015$ ($b = -0.065$) for the AR simulations, indicating that AR simulations reached a given precision about four times faster. More details on the performed numerical experiments can be found in Supplemental Material [31].

In conclusion, we proposed a novel approach to particle simulations which relies on a new, adaptively restrained Hamiltonian that switches positional degrees of freedom on and off during simulations, while letting momenta evolve. We have shown how adaptively restrained simulations allow us to smoothly trade between precision and speed, or compute static equilibrium properties faster.

We believe our approach may be extended in numerous directions, since the inverse mass matrix $\Phi(\mathbf{q}, \mathbf{p})$ can be chosen according to the specific needs of different types of simulations. Moreover, with any such matrix, ARPS can be combined with numerous existing methods, such as fast algorithms to compute long-range interactions [11,36] (although incremental versions may have to be designed), as well as techniques aimed at accelerating sampling [10,37,38]. We also want to explore several theoretical aspects of ARPS: the choice of thresholds, their influence on the system's dynamical properties, connections to Monte Carlo methods, etc.

We would like to thank Claude Le Bris, Frederic Legoll, Tony Lelievre, Mathias Rousset, and Gabriel Stoltz for numerous fruitful discussions and invaluable advice on mathematical and numerical aspects, Jean-François Scariot and Jean-Denis Séméria for their help with numerical tests, and the anonymous reviewers for helping us improve the manuscript.

[1] R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (Institute of Physics, Bristol, 1992).
[2] J. Monaghan, Rep. Prog. Phys. **68**, 1703 (2005).
[3] O. Etzmuss, J. Gross, and W. Strasser, IEEE Trans. Vis. Comput. Graphics **9**, 538 (2003).
[4] J. Izaguirre, S. Reich, and R. Skeel, J. Chem. Phys. **110**, 9853 (1999).
[5] C. Bennett, J. Comput. Phys. **19**, 267 (1975).
[6] K. Feenstra, B. Hess, and H. Berendsen, J. Comput. Chem. **20**, 786 (1999).
[7] M. Preto and S. Tremaine, Astron. J. **118**, 2532 (1999).
[8] F. Rao and M. Spichty, J. Comput. Chem. **33**, 475 (2012).
[9] P. Plechac and M. Rousset, Multiscale Model. Simul. **8**, 498 (2010).
[10] A. Voter, Phys. Rev. Lett. **78**, 3908 (1997).
[11] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
[12] J. Barnes and P. Hut, Nature (London) **324**, 446 (1986).
[13] P. B. Callahan and S. R. Kosaraju, J. Assoc. Comput. Mach. **42**, 67 (1995).
[14] M. Tuckerman, B. Berne, and G. Martyna, J. Chem. Phys. **97**, 1990 (1992).
[15] H. Lin and D. Truhlar, Theor. Chem. Acc. **117**, 185 (2007).
[16] O. M. Becker, A. D. MacKerell, Jr., B. Roux, and M. Watanabe, *Computational Biochemistry and Biophysics*, (Marcel-Dekker, Inc., New York, 2001).
[17] S. Reich, Numerical Algorithms **19**, 213 (1998).
[18] A. Gunaratne, Ph.D. thesis, Citeseer, 2006.
[19] M. Christen and W. van Gunsteren, J. Chem. Phys. **124**, 154 106 (2006).
[20] S. Marrink, H. Risselada, S. Yefimov, D. Tieleman, and A. De Vries, J. Phys. Chem. B **111**, 7812 (2007).
[21] *Coarse-Graining of Condensed Phase and Biomolecular Systems*, edited by G. A. Voth (CRC Press/Taylor and Francis Group, Boca Raton, 2009).
[22] A. Shih, A. Arkhipov, P. Freddolino, and K. Schulten, J. Phys. Chem. B **110**, 3674 (2006).
[23] M. Praprotnik, L. Site, and K. Kremer, Annu. Rev. Phys. Chem. **59**, 545 (2008).
[24] S. Nielsen, R. Bulo, P. Moore, and B. Ensing, Phys. Chem. Chem. Phys. **12**, 12 401 (2010).
[25] J. Park and A. Heyden, Mol. Simul. **35**, 962 (2009).
[26] B. Ensing, S. Nielsen, P. Moore, M. Klein, and M. Parrinello, J. Chem. Theory Comput. **3**, 1100 (2007).
[27] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, San Diego, 2002).
[28] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, J. Comput. Chem. **4**, 187 (1983).
[29] D. Brenner, Phys. Rev. B **42**, 9458 (1990).
[30] R. Rossi, M. Isorce, S. Morin, J. Flocard, K. Arumugam, S. Crouzy, M. Vivaudou, and S. Redon, Bioinformatics **23**, i408 (2007).
[31] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.109.190201 for theoretical proofs, details about algorithms, numerical experiments, and additional simulations.
[32] M. Bosson, S. Grudinin, X. Bouju, and S. Redon, J. Comput. Phys. **231**, 2581 (2012).
[33] E. Hairer, C. Lubich, and G. Wanner, in *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer Series in Computational Mathematics Vol. 31 (Springer-Verlag, Berlin, 2006), 2nd ed..
[34] T. Lelievre, G. Stoltz, and M. Rousset, *Free Energy Computations: A Mathematical Perspective* (Imperial College Press, London, 2010).
[35] A. Rahman, Phys. Rev. **136**, A405 (1964).
[36] T. Darden, D. York, and L. Pedersen, J. Chem. Phys. **98**, 10 089 (1993).
[37] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).
[38] A. Roitberg and R. Elber, J. Chem. Phys. **95**, 9277 (1991).