



OPEN

Fast, approximation-free molecular simulation of the SPC/Fw water model using non-reversible Markov chains

Philipp Höllmer¹, A. C. Maggs² & Werner Krauth^{3,4}✉

In a world made of atoms, computer simulations of molecular systems such as proteins in water play an enormous role in science. Software packages for molecular simulation have been developed for decades. They all discretize Hamilton's equations of motion and treat long-range potentials through cutoffs or discretization of reciprocal space. This introduces severe approximations and artifacts that must be controlled algorithmically. Here, we bring to fruition a paradigm for molecular simulation that relies on modern concepts in statistics to explore the thermodynamic equilibrium with an exact and efficient non-reversible Markov process. It is free of all discretizations, approximations, and cutoffs. We explicitly demonstrate that this approach reaches a break-even point with traditional molecular simulation performed at high precision, but without any of its approximations. We stress the potential of our paradigm for crucial applications in biophysics and other fields, and as a practical approach to molecular simulation. We set out a strategy to reach our goal of rigorous molecular simulation.

Keywords Molecular simulation, Non-reversible Markov chains, Monte Carlo methods, Long-range interactions

The fact that all matter consists of atoms was described by R. P. Feynman as the greatest insight of science¹. The consequence that matter can be modeled on a computer by following the motion of its atoms leads to the founding paradigm of molecular simulation. It tracks the dynamics and explores the thermodynamic equilibrium of complex molecular systems, for example, a peptide in an explicit water solution with tens of thousands of atoms, all interacting through classical empirical potentials². Molecular simulation is of enormous importance to numerous fields ranging from biology and physics to engineering^{3,4}. Powerful computer packages have been developed over decades. They all implement the molecular-dynamics approach^{5–9}, that is, compute the forces on all atoms at discretized time steps and then update the atomic positions and velocities to integrate Hamilton's equations of motion. This approach may yield static and dynamic properties both in non-equilibrium and in thermodynamic equilibrium.

A voluminous literature is dedicated to the analysis and control of time-discretization errors in molecular-dynamics simulations (see, e.g., Ref.¹⁰). Thermostats, understood as “necessary evils”¹¹, mimic the effect of a coupled thermal reservoir and, in a symptomatic but non-curative treatment, hide the accumulated errors. The limiting factor in molecular dynamics is the computation of forces. The Lennard-Jones interaction is typically cut off beyond a certain distance so that only a few neighbors exert their force on any given atom. This cutoff (or a discretization of reciprocal space akin to the approximate mesh-based Ewald methods^{12–16}) has long appeared as a necessity to increase computational efficiency. It is, however, artificial. On the one hand, its long-range nature translates the physics of the London dispersion force¹⁷. On the other hand, the phase diagram of the model depends strongly on the cutoff^{18,19}, so that results at different cutoffs have, in principle, to be extrapolated carefully. In the Lennard-Jones liquid, the surface tension and the Tolman length depend on the cutoff^{20,21}. The long-range nature of the Coulomb potential, which must be preserved¹¹, is usually treated through fast

¹Physikalisches Institut and Bethe Center for Theoretical Physics, University of Bonn, Nussallee 12, 53115 Bonn, Germany. ²CNRS UMR7083, ESPCI Paris, Université PSL, 10 Rue Vauquelin, 75005 Paris, France. ³Laboratoire de Physique de l'École normale supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université de Paris-Cité, Paris, France. ⁴Rudolf Peierls Centre for Theoretical Physics, University of Oxford, Oxford, UK. ✉email: werner.krauth@physics.ox.ac.uk; werner.krauth@ens.fr

mesh-based Ewald methods^{12–16} that solve the Poisson equation in discretized reciprocal space. The error of the computation and, thus, of the sampling of configurations, depends on a target-precision parameter that, in principle, calls for a costly extrapolation in order to be eliminated. Beyond their inherent approximations, cutoffs, discretizations of reciprocal space, and thermostats can even introduce unphysical artifacts in molecular-dynamics simulations^{11,12,22} that complex algorithms aim to keep under control.

Our alternative paradigm for the molecular simulation of static properties in thermodynamic equilibrium is based on several modern concepts in statistics^{23,24}, namely the factorized Metropolis filter²⁵, the concept of thinning²⁶ as expressed in the cell-veto algorithm²⁷, that of continuous-time piecewise-deterministic Markov processes²⁸ as realized in the event-chain Monte Carlo algorithm^{25,29,30} and, finally, Walker's method of aliases³¹ that can be adapted to the fast sampling of long-range interactions^{27,32}. The paradigm is rigorously exact from the start and, by construction, strictly samples the canonical ensemble without thermostats. Deterministic trajectories of atoms in continuous Monte-Carlo time follow an ordinary differential equation that can be integrated analytically. Trajectories are interrupted by *events* at random times. This non-reversible piecewise-deterministic Markov process violates the detailed-balance condition normally associated with thermal equilibrium, but still samples the Boltzmann distribution at each moment of its continuous-time evolution.

A particular algorithm that generates non-reversible Markov processes, event-chain Monte Carlo, has led to spectacular speedups of local Markov chains in statistical physics³³. In molecular systems, short- and long-range potentials are handled without any cutoffs or discretizations and, as we show in this paper, with competitive efficiency. The Boltzmann weight $\pi = \exp(-\beta U)$ (with β the inverse temperature and U the potential) is expressed as a factorized product $\pi = \prod_M \exp(-\beta U_M)$ of statistically independent factors M with factor potentials U_M with $\sum_M U_M = U$ that each depend only on a small subensemble of atoms²⁵. Every factor stochastically generates a candidate event time when the piecewise-deterministic motion must be interrupted. The minimum of these times triggers an event, and determines the initial conditions for the next piece. The total potential U and the corresponding forces never need evaluating, yet the stationary state is rigorously the Boltzmann distribution. This circumvents the problem of the inaccurate evaluation of the long-range forces between all atoms in the molecular-dynamics approach that is nowadays mostly used even for static properties in thermodynamic equilibrium. In the context of molecular simulation, the paradigm of non-reversible and event-driven Markov processes was up to now a mere theoretical possibility that converge only for the tiniest molecular systems^{23,24}.

The present paper is based on essential new developments in the event-driven paradigm. In particular, we generalize a particular non-reversible Markov-chain Monte Carlo algorithm for hard spheres, Newtonian event-chain Monte Carlo³⁰, to the factor potentials of molecular simulation. We implement this modified algorithm in demonstration software for N flexible SPC/Fw water molecules³⁴ interacting with the long-range Coulomb potential. As a benchmark, we concentrate on sampling the electric polarization³⁵ (the electric dipole moment of the simulation cell). Fluctuations in this quantity determine the dielectric properties of water. For large system sizes, we show that our demonstration software confirms the theoretical expectation²³ that the non-reversible Markov process decorrelates the polarization in a computer time that scales as $N \log N$, similar to state-of-the-art mesh-based Ewald methods in the molecular-dynamics approach¹³ but without a prefactor that diverges with the target precision for the force evaluations. Moreover, the generalized Newtonian event-chain Monte Carlo algorithm greatly reduces autocorrelation times compared to the previously considered event-chain Monte Carlo variant²³. This allows us to actually equilibrate large molecular systems. We show that our non-reversible algorithm overcomes the slow diffusive dynamics of reversible Monte Carlo algorithms, while the factorization does not penalize the dynamics in comparison to the molecular-dynamics approach. In terms of computer time, our approximation-free code reaches a break-even point with respect to a standard molecular-dynamics code that is run at a different target precisions for the long-range Coulomb interaction below machine precision. The considerable performance difference between event-chain Monte Carlo variants evidences the greater algorithmic freedom in our non-reversible Markov-process approach compared to the molecular-dynamics approach. Furthermore, we point out how our demonstration software can be improved in the future. Practical molecular simulations based on our paradigm are within reach, and we set out an interdisciplinary research strategy for sampling the Boltzmann distribution without any bias and for using non-reversible Markov chains as a gold standard for molecular simulation, in particular in the presence of long-range interactions.

Results

Modern-statistics paradigm for SPC/Fw water

In a molecular system with long-range interactions, the force on an atom depends on the position of all other atoms, rendering its evaluation tedious unless one introduces cutoffs or one discretizes reciprocal space. In contrast, we implement a piecewise-deterministic Markov process through the event-chain Monte Carlo algorithm, where a single atom moves with constant velocity at any given moment^{25,29,36}. The deterministic motion of this atom is interrupted by an event that stops it and sets off a similar motion of a new atom. Factors M in the SPC/Fw water model describe O–H bonds, the bending of H–O–H opening angles, O–O intermolecular Lennard-Jones interactions, and the Coulomb interaction between two water molecules (see Methods section). In a non-homogeneous Poisson process, each factor M proposes an independent *candidate event time* where an exponentially distributed random number is equal to the integrated *factor event rate*, that is, the cumulative increments of the factor potential U_M induced by the moving atom [see Eqs. (1) and (2) in the Methods section]. The minimum over all the candidate event times then realizes the next event, and motion is transferred to another atom contributing to the corresponding factor potential U_M . This succession of events takes place in continuous Monte Carlo time, and the Boltzmann distribution is sampled at all times^{28,36}. In comparison to the usual Metropolis algorithm³⁷, our formulation replaces rejections by transfers of motion.

In our implementation of the non-reversible Markov-chain paradigm in the JELLYFYSH application²⁴, positions and velocities of atoms define the *global state* of the physical system. To impose coherency of the physical system, the global state is accessed only through a central *mediator*³⁸ that dispatches physically independent computations of candidate events to *event handlers*. A *scheduler* weeds through candidate events. It identifies the unique event and the corresponding event handler that provides the subsequent transfer of motion, leading to an update of the global state (see Fig. 1). The event handlers within the mediator architecture mirror the statistical independence of the factors composing the physical system. This allows us to compose complex interactions in a transparent and independent manner. The pseudocode in Algorithm 1 in the Methods section summarizes the general implementation of an event computation within JELLYFYSH (including the cell-veto algorithm that is described in the next section).

A number of inequivalent options have been constructed for the update of active particles within event-chain Monte Carlo^{23,39}. Similar flexibility is possible in the updating of velocities^{29,30,40,41}, as well as in parallelizing the algorithm^{24,42}. The original straight variant is most efficient for simple hard disks^{29,33}, but is completely inappropriate for hard-disk dipoles⁴¹ and water molecules⁴³ that require the exploration of internal rotational degrees of freedom. In this paper, we generalize Newtonian event-chain Monte Carlo³⁰ from systems akin to the hard-disk system to molecular systems (see Methods section). It requires no fine-tuning and again exactly samples the canonical Boltzmann distribution. For the SPC/Fw water model, this paper shows that generalized Newtonian event-chain Monte Carlo is much more efficient than the fine-tuned straight variant. Piecewise-deterministic Markov processes offer an even wider choice of options for the management of events, the selection of factors, and the piecewise-deterministic trajectories, that may well apply to molecular simulation in the approximation-free non-reversible Markov-chain framework and may achieve further unforeseeable speedups.

Implementation for SPC/Fw water: Cell-veto—Fibonacci sphere

In the event-driven implementation of our method (see Fig. 1), a single atom moves among the other atoms and molecules, so that $O(N)$ long-range factors are changing with time and, in principle, yield independent events that would require sorting and managing. The cell-veto algorithm, as discussed below, allows one to bundle most of the long-range Coulomb and Lennard-Jones factors and, in the SPC/Fw water model, the mediator interacts

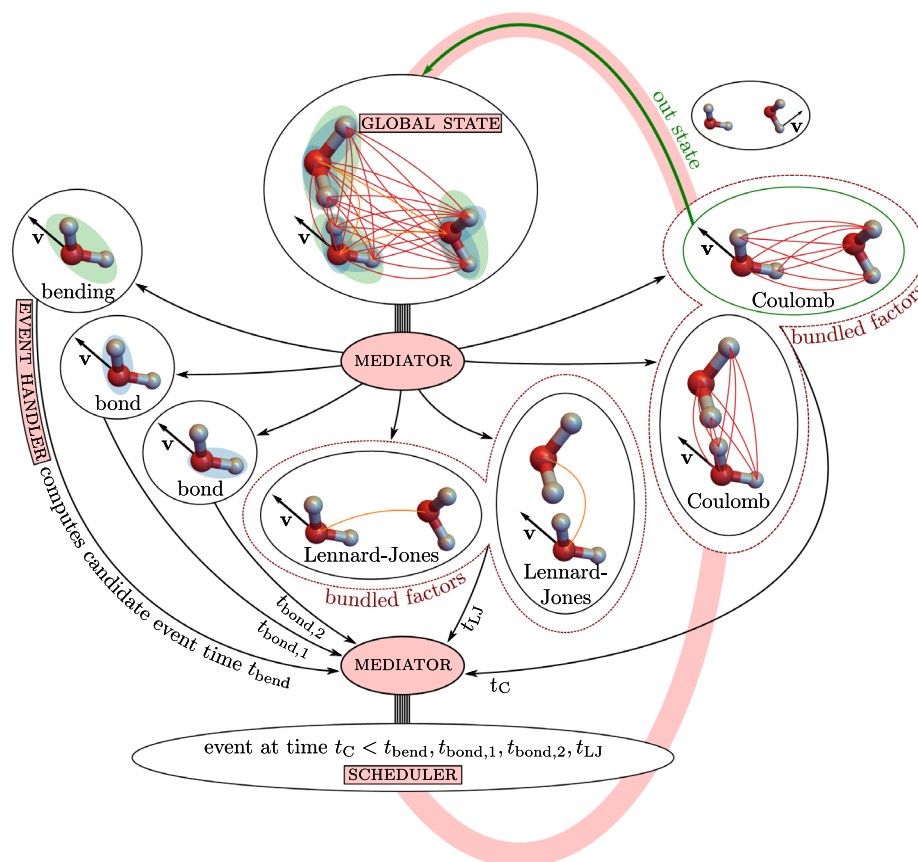


Figure 1. JELLYFYSH implementation of our Markov-chain paradigm. The mediator splits the *global state* into statistically independent factors. Factors communicate independent candidate event times, the earliest of which defines the next event. Factors for long-range interactions are bundled using the cell-veto algorithm, so that the number of *event handlers* remains limited. Candidate event times are collected by the mediator and then treated in the *scheduler*. The factor triggering the event then updates the global state, again *via* the mediator.

with only ~ 50 event handlers that propose candidate event times to the scheduler (see Fig. 2a). This number contains the two event handlers for the bundled long-range factors. In addition, it contains the event handlers for the bond and bending factors, and for the Coulomb and Lennard-Jones factors between nearby water molecules that were specifically excluded to minimize computer time (see Methods section). The bundling allows the processing of each event in constant computer time (for large N) while treating the long-range interactions without approximations.

In our context of the SPC/Fw water model, the cell-veto algorithm²⁷ upper-bounds the factor event rates for pairs of molecules interacting with the Coulomb potential by precomputed, time-independent bounds for these molecules somewhere within a pair of cells (see Fig. 2b). The full set of these *cell bounds* corresponds to the set of bundled factors of the long-range interaction. Walker's method of aliases³¹ conserves the cell bounds in a *Walker table*. In the event-driven evolution of the piecewise-deterministic Markov process, the set of cell bounds in the Walker table provides a single candidate event time for the entire set of bundled factors, and Walker's method samples an associated single cell bound—and thus a single associated factor—with constant algorithmic complexity. The overestimation of the factor event rate by the cell bound is exactly corrected in a procedure akin to the thinning of non-homogeneous Poisson processes²⁶ by confirming the transfer of motion in the event (see Fig. 2d and Algorithm 1). This thinning is performed with the actual positions of the atoms and the actual velocity of the moving atom, leading to an exact treatment of the long-range interaction that is independent of the set of cell bounds (see Ref.⁴⁴ for a simple example on how the thinning procedure exactly corrects for the overestimation of the cell bounds with respect to the underlying factor event rates).

The factor event rates, and hence the cell bounds, depend on the velocity of the moving atom. In all previous applications of the cell-veto algorithm in conjunction with the original straight event-chain Monte Carlo variant, there was only a small number of possible velocities and a Walker table was built for each of them^{23,24,27}. In order to use the cell-veto algorithm with our generalized Newtonian event-chain Monte Carlo variant, we discretize its continuous velocity space for the precomputation of the cell bounds. We build separate Walker tables for multiple directions of the velocity of the moving atom corresponding to *Fibonacci vectors* on the unit sphere (see Fig. 2c and Methods section). During the simulation, the actual velocity of the moving atom is mapped to the closest Fibonacci vector. The thinning procedure again exactly corrects for this discretization of velocity space.

Benchmarking for SPC/Fw water

For our benchmark, we release a new major version of the JELLYFYSH application²⁴ that implements generalized Newtonian event-chain Monte Carlo and, consequently, the cell-veto algorithm with a discretized velocity space. Only the new version actually succeeds in equilibrating large configurations of N SPC/Fw water molecules in a periodic box at standard density and temperature. We implement long-range molecular Coulomb factors with Walker tables that we also adopt for the Lennard-Jones interaction (see Methods section for details). We find that generalized Newtonian event-chain Monte Carlo, for large N , requires a computer time per event that remains constant (see Fig. 3a). A large number of unconfirmed events stems from the overestimated cell bounds which, e.g., do not account for the relative orientation of molecules (see inset of Fig. 3a). Since every unconfirmed event requires a high-precision evaluation of a long-range interaction between two water molecules, reducing this number will much reduce computer times. While the computer time per event is constant, the number of events per Ångström (that is, per unit Monte-Carlo time) increases logarithmically with N for the Coulomb factors

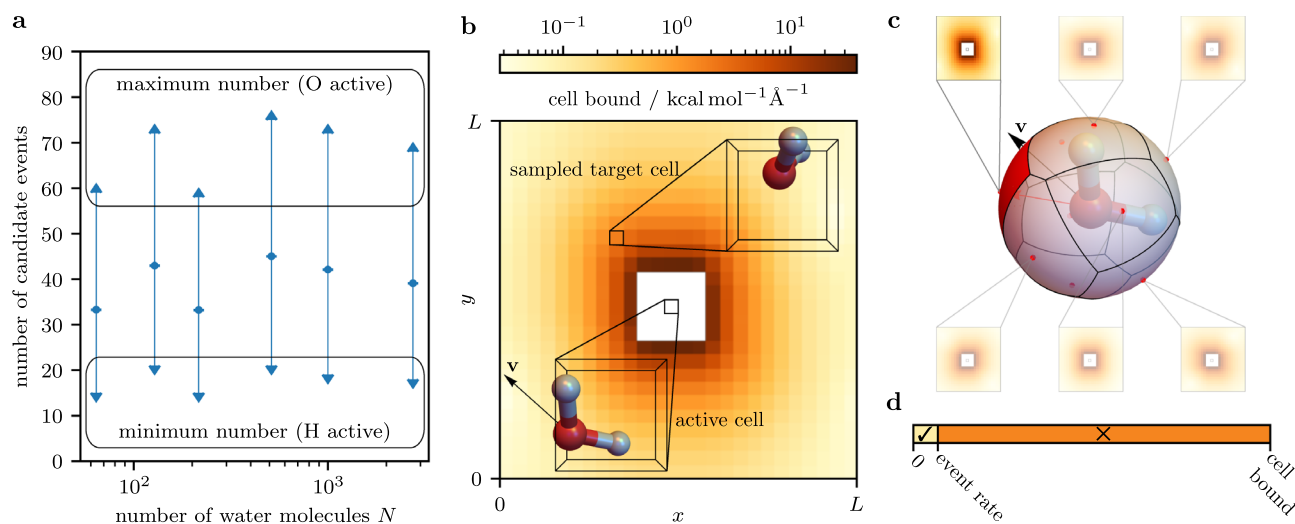


Figure 2. Long-range interactions with constant computer time per event. **(a)** The number of candidate events (event handlers) is constant for increasing system sizes. **(b)** Walker table from which a target cell is sampled according to its cell bound with respect to the active cell containing the moving atom. **(c)** Different Walker tables for Fibonacci vectors on the unit sphere. The active atom obtains cell bounds from the nearest vector. **(d)** The thinning procedure confirms (✓) or rejects (X) the event using the actual factor event rate of the molecules in the active and target cell.

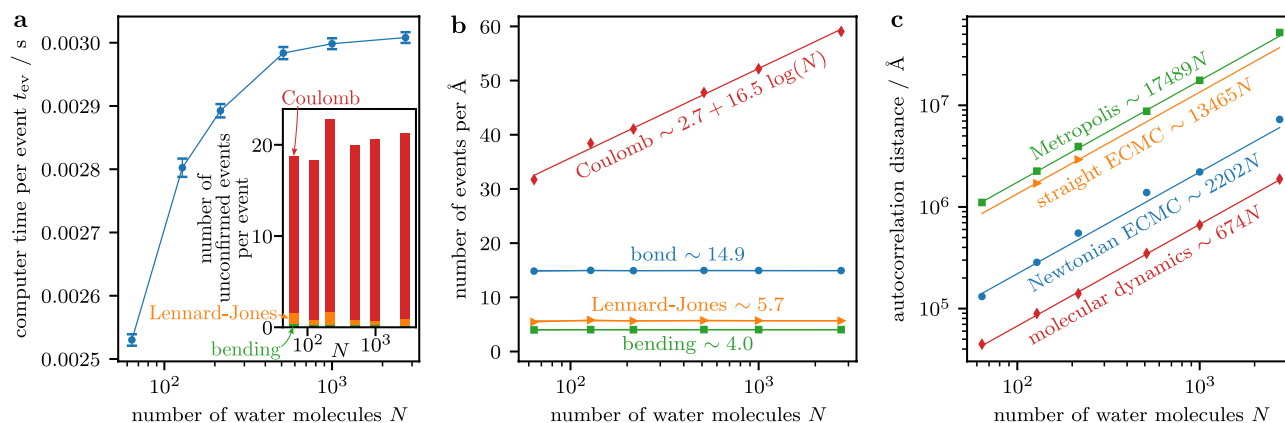


Figure 3. Event rates and decorrelation in the SPC/Fw water model. **(a)** Computer time per event in JELLYFYSH. Inset: Number of unconfirmed events per event for different factor types. **(b)** Event rate in JELLYFYSH for different factor types. **(c)** Distance to decorrelate the polarization for different sampling algorithms (for molecular dynamics: sum over the average displacements of all atoms per time step).

(see Fig. 3b), as predicted by theory²³. In summary, our approach requires a computer time scaling as $N \log N$ to advance N water molecules by a constant distance. This matches the complexity of mesh-based Ewald methods in the molecular-dynamics approach¹³, but without their slowdown as the target precision is increased because JELLYFYSH treated long-range interactions exactly to begin with.

For concreteness, we compare the decorrelation of the polarization of the water system within JELLYFYSH to molecular-dynamics simulations within the LAMMPS software⁸ on a single processor with default parameters and a 1 fs time step (see Methods section). To decorrelate this local quantity, both LAMMPS and JELLYFYSH must move the atoms of any water molecule by a characteristic distance (see Fig. 3c and Methods section for details). This characteristic distance lacks a clear physical interpretation for the Metropolis and event-chain Monte Carlo algorithms because of their *ad-hoc* dynamics for exploring sample space. However, it provides a fair measure of the distance in sample space that separates independent samples (“lower” means “faster”). Different variants of event-chain Monte Carlo vary in their efficiency, and our generalized Newtonian variant is an order of magnitude faster than the original straight variant as the characteristic distance is reduced from 13465 Å to 2202 Å. In comparison to molecular dynamics with the smallest characteristic distance of 674 Å, the non-reversible Markov process only moves a single atom at any point in time which may explain the slightly larger characteristic distance of generalized Newtonian event-chain Monte Carlo. The reversible Metropolis algorithm with analogous single-atom moves (that, in principle, also reaches an $N \log N$ scaling⁴⁵ by using a recent variant of the fast multipole method⁴⁶), as implemented in the DL_MONTE software package⁴⁷, is clearly inferior to our non-reversible methods with a characteristic distance of 17489 Å.

In the molecular-dynamics simulations that we benchmark our approach against, we use a state-of-the-art mesh-based Ewald method to treat the long-range Coulomb interaction. Its implementation in LAMMPS sets a target accuracy based on analytic error estimates obtained from a specific charge distribution^{15,48,49}. To accelerate the evaluation of the reciprocal sum in the Ewald method, the charges of atoms (which live in continuous space) are mapped onto a grid using an interpolation scheme. Finer grids and higher interpolation orders yield higher target accuracies. For various system sizes, we estimate the required computer times for different target accuracies by changing the grid spacing, using the LAMMPS defaults. Naturally, higher target accuracies require more computer time (see Fig. 4a). Because of the underlying factorization of the Boltzmann distribution, event-chain Monte Carlo is rigorously approximation-free. It only explicitly considers the Coulomb interaction between two SPC/Fw water molecules (mostly when candidate events from the cell-veto algorithm are confirmed). Our implementation in JELLYFYSH uses the historic Ewald summation in continuous space for this interaction between a small number of charges that stays constant with increasing system size. We tune it to machine precision without any assumptions on the global charge distribution. In comparison to the molecular-dynamics approach at different target accuracies for the long-range Coulomb interaction as implemented in LAMMPS, our non-reversible piecewise-deterministic Markov process, specifically the generalized Newtonian event-chain Monte Carlo as implemented in JELLYFYSH, reaches the break-even point well below machine precision (see Fig. 4b).

Discussion

In this paper, we benchmarked an implementation of a modern-statistics paradigm for molecular simulations of static properties in thermodynamic equilibrium in the standard SPC/Fw water model. The time dependence of the corresponding Markov process differs from the physical dynamics yet it exactly approaches thermal equilibrium on time scales that are potentially faster than in nature. Its remarkable efficiency (that we expressed as an $N \log N$ computer time to decorrelate a local observable, the polarization) is rooted in three paradoxes. First, the Markov process is non-reversible (that is, effectively out-of-equilibrium), yet its steady state coincides with the equilibrium Boltzmann distribution⁵⁰. In contrast to standard Monte Carlo algorithms that satisfy the detailed-balance condition and only move diffusively, it features finite probability flows, making it capable of moving ballistically. This has already led to considerable speedups in a variety of fields ranging from physics

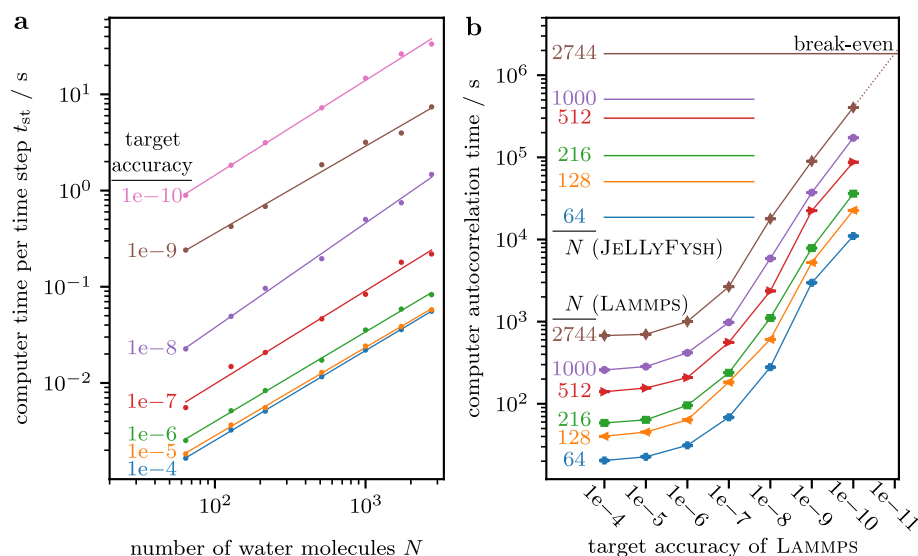


Figure 4. LAMMPS–JELLYFYSH benchmark for the SPC/Fw water model. **(a)** Computer time per step of LAMMPS for different target accuracies of its particle–particle particle–mesh solver. **(b)** Computer time used by LAMMPS to decorrelate the polarization depends on the target accuracy and the number of water molecules N . JELLYFYSH is exact up to machine precision. The break-even precision is indicated.

to statistics and machine learning (see, e.g., Refs.^{36,51–53}). In statistical mechanics, a decrease in the dynamical scaling exponents was demonstrated in a number of interacting particle models in one and higher dimensions, resulting in speedups with respect to reversible Markov chains and to molecular dynamics that diverge with the system size^{54,55}. It remains to be seen whether non-local observables as, for example, large-scale hydrodynamic modes, macroscopic conformations, and order parameters can similarly, in chemical physics, benefit from non-reversibility. The second paradox in our approach is that the Boltzmann distribution $\exp(-\beta U)$ is sampled without any approximation and with great efficiency although the total potential U and its derivatives, the forces, are never evaluated. This sidesteps all the problems with limited-precision calculations of energies and forces. The third paradox is the bundling of $\mathcal{O}(N)$ independent decisions to interrupt the straight-line trajectory of the piecewise-deterministic Markov process into an expression that can be evaluated in constant time. The Walker tables bundle a multitude of long-range Coulomb factors and of Lennard-Jones factors into two candidate events, one for long-range Coulomb, one for long-range Lennard-Jones, which allows us to handle a complex decision (a conjunction of $\mathcal{O}(N)$ factor-wise decisions of independent factor potentials U_M) in a few operations, even in the $N \rightarrow \infty$ limit.

Our software JELLYFYSH is openly available and fully functional, although it has only the status of demonstration software. It implements the interactions of SPC/Fw water, and provides all necessary configuration files to replicate the simulations of this paper out-of-the-box. In the Methods section, we further provide the detailed simulation protocol of event-chain Monte Carlo in JELLYFYSH. The flexible mediator-based architecture of JELLYFYSH allows for simple extensions to molecular systems beyond water molecules. Although it is still mostly written in the relatively slow Python language, the new major version that we have described in this paper is of practical use for molecular simulation. Our method is exact from the very beginning and our application becomes competitive with the traditional molecular-dynamics code LAMMPS at high intrinsic precision. Optimizing JELLYFYSH in the same way that LAMMPS was optimized over decades will greatly reduce the break-even point. In addition, future research will be able to concentrate on the most efficient ones among a large choice of cell bounds, factorizations, Fibonacci vectors and variants of the piecewise-deterministic Markov processes. The most efficient application to non-homogeneous systems and to non-homogeneous (for example, corrugated) boundary conditions will be other interesting directions of research. The parallelization of our method to a large number of active particles has for the moment been implemented only for hard-disk potentials⁴². The generalization to arbitrary potentials presents an outstanding challenge. Clearly, more interdisciplinary research from statistics to computational chemistry will clarify whether all of this provides a sufficiently strong basis for an alternative approach for practical molecular simulation.

With its guarantee for the unbiased sampling of the Boltzmann distributions, our paradigm may serve as a gold standard for molecular simulation in general, capable of identifying artifacts and approximations that may not have been totally eliminated through the symptomatic algorithmic approach in molecular-dynamics simulations. Furthermore, given the greater algorithmic freedom for the Markov-chain approach than for the molecular-dynamics approach, it may actually become faster than molecular dynamics to explore thermodynamic equilibrium. Several orders of magnitude in numerical speed can certainly be gained by re-engineering our software, which would then be able to tackle the peptide-in-water benchmark problem² that has had a major influence over the last decade. The great simplicity of our approach and its present implementation in the JELLYFYSH software may well facilitate further developments. Ultimately, our paradigm may yield independent equilibrium

samples that then serve, in synergy with long-established molecular-dynamics software, as starting configurations for parallel molecular dynamics calculations in order to access high-precision dynamical correlation functions.

Methods

Generalized Newtonian event-chain Monte Carlo

Event-chain Monte Carlo is a family of local non-reversible Markov-chain Monte Carlo algorithms that implement piecewise-deterministic Markov processes. All variants are event-driven and move a single atom along a straight-line trajectory at any time. They differ only in their update of the moving atom and its velocity in the transfers of motion at event times. Straight event-chain Monte Carlo is the original variant and proved to be most efficient for the hard-disk model^{29,33}. Its generalization to the translationally invariant factor potentials of molecular systems^{23,25} is implemented in the version 1.0 of JELLYFYSH²⁴.

We develop version 2.0 of JELLYFYSH⁵⁶ for this paper. The pseudocode in Algorithm 1 provides a high-level summary of its general implementation of a piecewise-deterministic Markov process. This pseudocode complements the mathematical description in the following, and Fig. 5 in Ref.²⁴ that introduced the basic architecture of JELLYFYSH. Besides the straight variant, version 2.0 of JELLYFYSH also implements the generalized Newtonian event-chain Monte Carlo algorithm. Newtonian event-chain Monte Carlo was originally only formulated for the hard-sphere model that contains peculiar stepwise-changing two-body factor potentials³⁰. Among various event-chain Monte Carlo variants, it was shown to escape faster from sparse hard-disk packings⁵⁷ and to produce favorable rotation dynamics in tethered hard-disk dipoles⁴¹. Its superiority is confirmed for the rotation dynamics of SPC/Fw water molecules in this paper, but requires its generalization to smooth factor potentials that depend on an arbitrary number of atom positions.

```

procedure next_event in mediator
get moving atom  $i$  and its velocity  $\mathbf{v}_i$  from global state
for each bundle  $M_b$  of factors do # number of  $M_b \sim \text{constant}$ 
  if ( $i \in M_b$ ) then
    get candidate event time  $t_{ev, M_b}$  from event handler  $E_{M_b}$ 
    send  $t_{ev, M_b}$  to scheduler
for each non-bundled factor  $M_{nb}$  do # number of  $M_{nb} \sim \text{constant}$ 
  if ( $i \in M_{nb}$ ) then
    get candidate event time  $t_{ev, M_{nb}}$  from event handler  $E_{M_{nb}}$ 
    send  $t_{ev, M_{nb}}$  to scheduler
1 get smallest candidate event time  $t_{ev}$  and event handler  $E_{ev}$  from scheduler
if ( $E_{ev}$  treats bundle  $M_b$  of factors) then
  get single sampled factor  $M_{ev} \in M_b$  from  $E_{ev}$  #  $\mathcal{O}(1)$  complexity
  # thinning leads to exact treatment of factors  $M_b$ 
  if ( $E_{ev}$  confirms event by thinning) then
    send new position  $\mathbf{x}_i \rightarrow \mathbf{x}_i + \mathbf{v}_i t_{ev}$  of  $i$  to global state
    get new moving atom  $j$  and velocity  $\mathbf{v}'$  from  $E_{ev}$ 
    send  $j$  and  $\mathbf{v}'$  to global state
    return # event realized
  else # event of factor  $M_{ev} \in M_b$  not confirmed
    # next candidate event time is always bigger than the old one
    get next candidate event time  $t_{ev, M_b}$  from event handler  $E_{M_b}$ 
    send  $t_{ev, M_b}$  to scheduler
    goto 1 # find smallest candidate event time again
else #  $E_{ev}$  treats non-bundled factor  $M_{nb} = M_{ev}$ 
  # thinning leads to exact treatment of factor  $M_{nb}$ 
  if ( $E_{ev}$  treats  $M_{ev}$  without overestimated factor event rate
  or  $E_{ev}$  confirms event by thinning) then
    send new position  $\mathbf{x}_i \rightarrow \mathbf{x}_i + \mathbf{v}_i t_{ev}$  of  $i$  to global state
    get new moving atom  $j$  and velocity  $\mathbf{v}'$  from  $E_{ev}$ 
    send  $j$  and  $\mathbf{v}'$  to global state
    return # event realized
  else # event of factor  $M_{nb} = M_{ev}$  not confirmed
    # next candidate event time is always bigger than the old one
    get next candidate event time  $t_{ev, M_{nb}}$  from event handler  $E_{M_{nb}}$ 
    send  $t_{ev, M_{nb}}$  to scheduler
    goto 1 # find smallest candidate event time again

```

Algorithm 1. Procedure that realizes the next event in the JELLYFYSH implementation. The *mediator* communicates with the *global state*, the *event handlers*, and the *scheduler*. Factors for long-range interactions are bundled to achieve a constant number of event handlers that compute candidate event times. Thinning exactly corrects for candidate event times that were computed based on overestimated factor event rates.

We follow the initial introduction of non-reversible event-chain Monte Carlo as a lifted continuous-time Markov process²⁵ (see also the formulation as a piecewise-deterministic Markov process in Ref.⁵⁸). Consider N_a atoms and let $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_{N_a}) \in \mathbb{R}^{3N_a}$ be the physical all-atom configuration that collects all three-dimensional atom positions $\mathbf{x}_i \in \mathbb{R}^3$ with $i \in \{1, \dots, N_a\}$. For generalized Newtonian event-chain Monte Carlo, the lifting framework^{59,60} (see also Refs³⁶ and⁶¹, Appendix A) extends the physical all-atom configuration \mathbf{x} to $(\mathbf{x}, \mathbf{v}, i)$, with auxiliary lifting variables that represent an all-atom velocity \mathbf{v} and an activity label i , respectively. The physical sample space $\mathbf{x} \in \Omega$ is extended to the lifted sample space $(\mathbf{x}, \mathbf{v}, i) \in \widehat{\Omega} = \Omega \times \mathcal{V}^{3N_a}(E_{\text{kin}}) \times \mathcal{N}$, where $\mathcal{N} = \{1, \dots, N_a\}$ is the set of atom indices, and $\mathcal{V}^{3N_a}(E_{\text{kin}}) = \{\mathbf{v} \in \mathbb{R}^{3N_a} : \mathbf{v}^T \mathbf{M} \mathbf{v} = 2E_{\text{kin}}\}$ with a positive-definite symmetric mass matrix \mathbf{M} and conserved total kinetic energy E_{kin} . The Markov process targets the

lifted stationary distribution $\hat{\pi}(\mathbf{x}, \mathbf{v}, i) = \pi(\mathbf{x}) \times \mu_{\mathcal{V}}(\mathbf{v}) \times \mu_{\mathcal{N}}(i)$ that separates into the factorized Boltzmann distribution $\pi(\mathbf{x}) \propto \exp[-\beta U(\mathbf{x})] = \prod_M \exp[-\beta U_M(\mathbf{x})]$, and the uniform distributions $\mu_{\mathcal{V}}(\mathbf{v})$ on \mathcal{V}^{3N_a} , and $\mu_{\mathcal{N}}(i)$ on \mathcal{N} . Since the kinetic energy is fixed throughout a simulation of generalized event-chain Monte Carlo, its kinetic-energy distribution does not coincide with that of the canonical ensemble. The all-atom “velocity” lacks a physical meaning.

Given a lifted configuration $(\mathbf{x}(t_0), \mathbf{v}(t_0), i)$ at time t_0 , Newtonian event-chain Monte Carlo continuously moves the single active atom i starting from its position $\mathbf{x}_i(t_0)$ with its constant velocity $\mathbf{v}_i = \mathbf{v}_i(t_0)$ up to an event at time $t_{ev} > t_0$, which interrupts the motion: $\mathbf{x}_i(t) = \mathbf{x}_i(t_0) + \mathbf{v}_i(t - t_0)$ for $t_0 \leq t < t_{ev}$. The velocities \mathbf{v}_j of all other atoms $j \neq i$ in the all-atom velocity \mathbf{v} are, at this point, hypothetical, that is, mere labels. The time-dependent event rate $\lambda(t)$, that is, the probability density to interrupt the piecewise-deterministic motion of the active atom i , is given by a sum of factor event rates $\lambda_M(t)$:

$$\lambda(t) = \sum_M \lambda_M(t) = \sum_M \beta \max [0, \nabla_{\mathbf{x}_i} U_M(\mathbf{x}(t)) \cdot \mathbf{v}_i]. \quad (1)$$

All factors M can be considered as statistically independent. Each of them stochastically generates a candidate event time $t_{ev,M}$ in an inhomogeneous Poisson process based on its factor event rate $\lambda_M(t) \geq 0$ (that is only nonzero for factor potentials U_M that are actually changed by the motion of the active atom):

$$\text{ran}_M(0, 1) = \exp \left[- \int_{t_0}^{t_{ev,M}} \lambda_M(t) dt \right]. \quad (2)$$

Here, $\text{ran}_M(0, 1)$ is a uniformly distributed random number between 0 and 1 that is drawn separately for each factor M . In Eq. (2), the cumulative increments of the factor potential U_M under the motion of the active atom with its velocity \mathbf{v}_i since t_0 are equal to an exponentially distributed random number with mean $1/\beta$ at the time $t_{ev,M}$. If the computation of $t_{ev,M}$ from Eq. (2) is tedious or even impossible, a Poisson thinning procedure replaces the factor event rate $\lambda_M(t)$ by an upper bound $\tilde{\lambda}_M(t)$ that satisfies $\tilde{\lambda}_M(t) \geq \lambda_M(t)$ for all t ^{26,36}. Superfluous events from the increased factor event rate are thinned out by first confirming an event at the sampled candidate event time $t_{ev,M}$ with probability $\lambda_M(t_{ev,M})/\tilde{\lambda}_M(t_{ev,M}) \leq 1$.

At the (confirmed) event time $t_{ev} = \min_M t_{ev,M}$, the motion of the active atom i is interrupted at the lifted configuration $(\mathbf{x} = \mathbf{x}(t_{ev}), \mathbf{v}, i)$. An event changes the lifting variables and sets the initial lifted configuration $(\mathbf{x}, \mathbf{v}', j)$ for the next piece in the piecewise-deterministic Markov process. The event is realized by a unique event factor $M_{ev} = \arg \min_M t_{ev,M}$. Generalized Newtonian event-chain Monte Carlo proposes an update of the all-atom velocity $\mathbf{v} \rightarrow \mathbf{v}'$ at time t_{ev} with a “force kick” in the direction of the gradient of the event-factor potential:

$$\mathbf{v}' = \mathbf{v} - 2 \frac{\mathbf{v} \cdot \nabla_{\mathbf{x}} U_{M_{ev}}}{(\nabla_{\mathbf{x}} U_{M_{ev}})^T \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_{M_{ev}}} \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_{M_{ev}}. \quad (3)$$

This force kick leaves the kinetic energy $2E_{kin} = \mathbf{v}^T \mathbf{M} \mathbf{v}$ invariant and applying it twice yields the original \mathbf{v} ⁶². The event-factor potential only acts on a small subensemble $k \in M_{ev}$ of atoms, and thus only the velocities \mathbf{v}_k of these contributing atoms are possibly modified. For the simple choice of an identity mass matrix $\mathbf{M} = \mathbf{I}$, Eq. (3) may be written as

$$\mathbf{v}'_k = \mathbf{v}_k - 2 \frac{\sum_{j \in M_{ev}} \mathbf{v}_j \cdot \nabla_{\mathbf{x}_j} U_{M_{ev}}}{\sum_{j \in M_{ev}} |\nabla_{\mathbf{x}_j} U_{M_{ev}}|^2} \nabla_{\mathbf{x}_k} U_{M_{ev}}. \quad (4)$$

In generalized Newtonian event-chain Monte Carlo, only a single active atom moves with its velocity at any time. At an event, it must choose the next active atom $k \in M_{ev}$ from the event factor. It also chooses to either change the velocity $\mathbf{v} \rightarrow \mathbf{v}'$ of all contributing atoms according to Eq. (3), or to keep the all-atom velocity constant $\mathbf{v} \rightarrow \mathbf{v}$. Generalized Newtonian event-chain Monte Carlo offers two schemes to treat events. The *Newtonian-general* scheme applies to general mass matrices \mathbf{M} and factors that depend on an arbitrary number of atoms. The *Newtonian-pair* scheme only applies to distance-dependent pair potentials and an identity mass matrix $\mathbf{M} = \mathbf{I}$.

In the Newtonian-general scheme, the probability for each possible choice (k, \mathbf{v}') or (k, \mathbf{v}) of the lifting variables is given by $\max[0, -\nabla_{\mathbf{x}_k} U_{M_{ev}} \cdot \mathbf{v}'_k]/C$ or $\max[0, -\nabla_{\mathbf{x}_k} U_{M_{ev}} \cdot \mathbf{v}_k]/C$, respectively. Here, C is a common normalization factor. If the force kick is not applied and \mathbf{v} stays constant, the active atom always changes because the factor event rate was positive at the time of the event [see Eqs. (1) and (2)]. Likewise, if the event-factor potential only depends on a single atom (as for external potentials), the force kick is always applied.

In the Newtonian-pair scheme for distance-dependent pair potentials between two atoms a and b , $U_{M_{ev}} = U_{M_{ev}}(|\mathbf{x}_a - \mathbf{x}_b|)$, and an identity mass matrix $\mathbf{M} = \mathbf{I}$, we alternatively exploit the inherent translational invariance to treat the event deterministically. We always apply the force kick in Eq. (4) to modify both \mathbf{v}_a and \mathbf{v}_b , and always change the active atom within the event factor. In this case, Eq. (4) is equivalent to an elastic Newtonian collision between two particles of equal mass. This alternative scheme recovers the original formulation of Newtonian event-chain Monte Carlo for the hard-sphere model³⁰.

Samples of all-atom configurations \mathbf{x} that can be used to compute observables are taken at periodic time intervals³⁰. Furthermore, Newtonian event-chain Monte Carlo requires resampling of the all-atom velocity \mathbf{v} and the active atom i at periodic time intervals τ_{chain} in order to be irreducible⁴¹. We sample \mathbf{v} and i from their respective stationary distributions $\mu_{\mathcal{V}}(\mathbf{v})$ and $\mu_{\mathcal{N}}(i)$. For the all-atom velocity \mathbf{v} , this is equivalent to sampling a multivariate Gaussian distribution with mean $\boldsymbol{\mu} = \mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma} = \mathbf{M}^{-1}$, followed by corrections

of the sampled total velocity so that $\mathbf{1} \cdot \mathbf{v}=0$ and $\mathbf{v}^T \mathbf{M} \mathbf{v}=2E_{\text{kin}}$. The activity label i is sampled uniformly from the set $\mathcal{N} = \{1 \dots N_a\}$.

Proof of correctness of generalized Newtonian event-chain Monte Carlo

In order to prove the correctness of the Newtonian event-chain Monte Carlo algorithm for general smooth interactions, we show that the global balance condition is satisfied. The total probability flow into any lifted configuration $(\mathbf{x}, \mathbf{v}, i) \in \widehat{\Omega}$ consists of a physical flow $\mathcal{F}^{\text{phys}}(\mathbf{x}, \mathbf{v}, i)$ and a lifting flow $\mathcal{F}^{\text{lift}}(\mathbf{x}, \mathbf{v}, i)$, and must equal its statistical weight $\hat{\pi}(\mathbf{x}, \mathbf{v}, i)$:

$$\hat{\pi}(\mathbf{x}, \mathbf{v}, i) = \mathcal{F}^{\text{phys}}(\mathbf{x}, \mathbf{v}, i) + \mathcal{F}^{\text{lift}}(\mathbf{x}, \mathbf{v}, i). \tag{5}$$

The physical flow into $(\mathbf{x}, \mathbf{v}, i)$ stems from the continuous movement of the active atom i that was not interrupted by an event with the event rate given in Eq. (1). With $\mathbf{x}' = (\mathbf{x}_1, \dots, \mathbf{x}_i - \mathbf{v}_i dt, \dots, \mathbf{x}_{N_a})$, we get

$$\begin{aligned} \mathcal{F}^{\text{phys}}(\mathbf{x}, \mathbf{v}, i) &= \hat{\pi}(\mathbf{x}', \mathbf{v}, i) \left\{ 1 - \beta \sum_M \max \left[0, \nabla_{\mathbf{x}'_i} U_M \cdot \mathbf{v}_i \right] \right\} \\ &= \hat{\pi}(\mathbf{x}, \mathbf{v}, i) \left\{ 1 - \beta \sum_M \max \left[0, -\nabla_{\mathbf{x}_i} U_M \cdot \mathbf{v}_i \right] \right\} \\ &= \hat{\pi}(\mathbf{x}, \mathbf{v}, i) + \sum_M \mathcal{F}_M^{\text{phys}}(\mathbf{x}, \mathbf{v}, i), \end{aligned} \tag{6}$$

where the second line uses the detailed-balance property of the factorized Metropolis filter that yields the event rates in Eq. (1) and allows treating all factors as being statistically independent²⁵. The lifting flow into $(\mathbf{x}, \mathbf{v}, i)$ stems from interrupted motions of lifted configurations $(\mathbf{x}, \mathbf{v}', j)$ with different lifting variables \mathbf{v}' and j but the same physical all-atom configuration \mathbf{x} :

$$\begin{aligned} \mathcal{F}^{\text{lift}}(\mathbf{x}, \mathbf{v}, i) &= \beta \sum_M \sum_{j \in \mathcal{N}} \int_{\mathcal{V}^{3N_a}(E_{\text{kin}})} d^{3N_a} \mathbf{v}' \Theta(\mathbf{x}, \mathbf{v}', j) \max \left[0, \nabla_{\mathbf{x}_j} U_M \cdot \mathbf{v}'_j \right] p_{(\mathbf{v}', j), (\mathbf{v}, i)}^M \\ &= \sum_M \mathcal{F}_M^{\text{lift}}(\mathbf{x}, \mathbf{v}, i). \end{aligned} \tag{7}$$

Here, $p_{(\mathbf{v}', j), (\mathbf{v}, i)}$ is the probability to change the lifting variables (\mathbf{v}', j) to (\mathbf{v}, i) . For every factor M , the lifting flow $\mathcal{F}_M^{\text{lift}}(\mathbf{x}, \mathbf{v}, i)$ in Eq. (7) exactly cancels the physical flow $\mathcal{F}_M^{\text{phys}}(\mathbf{x}, \mathbf{v}, i)$ in Eq. (6) so that the global-balance condition in Eq. (5) is satisfied. A trivial (inefficient) solution would be to simply invert the velocity of the active atom i in an event while keeping it active: $p_{(\mathbf{v}', j), (\mathbf{v}, i)}^M = \delta_{ij} \delta^{(3)}(\mathbf{v}_i - \mathbf{v}'_i)$.

We presented two schemes for the update of the lifting variables in an event of generalized Newtonian event-chain Monte Carlo. The first stochastic Newtonian-general scheme applies for a general number of atoms on the event factor. The second deterministic Newtonian-pair scheme exploits the translational invariance of distance-dependent pair potentials. For the Newtonian-general scheme, we get

$$\begin{aligned} p_{(\mathbf{v}', j), (\mathbf{v}, i)}^M &= \frac{\max \left[0, -\nabla_{\mathbf{x}_i} U_M \cdot \mathbf{v}_i \right]}{C} \left[\delta^{(3N_a)} \left(\mathbf{v} - \mathbf{v}' + 2 \frac{\mathbf{v}' \cdot \nabla_{\mathbf{x}} U_M}{(\nabla_{\mathbf{x}} U_M)^T \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M} \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M \right) + \delta^{(3N_a)}(\mathbf{v} - \mathbf{v}') \right] \\ &= \frac{\max \left[0, -\nabla_{\mathbf{x}_i} U_M \cdot \mathbf{v}_i \right]}{C} \left[\delta^{(3N_a)} \left(\mathbf{v}' - \mathbf{v} + 2 \frac{\mathbf{v} \cdot \nabla_{\mathbf{x}} U_M}{(\nabla_{\mathbf{x}} U_M)^T \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M} \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M \right) + \delta^{(3N_a)}(\mathbf{v}' - \mathbf{v}) \right], \end{aligned} \tag{8}$$

where the first term applies the force kick, while the second does not. From the necessary condition

$$1 = \sum_{k \in \mathcal{N}} \int_{\mathcal{V}^{3N_a}(E_{\text{kin}})} d^{3N_a} \mathbf{v} p_{(\mathbf{v}', j), (\mathbf{v}, k)}^M, \tag{9}$$

it follows that

$$\begin{aligned}
 C &= \sum_{k \in \mathcal{N}} \left\{ \max \left[0, -\nabla_{\mathbf{x}_k} U_M \cdot \left(\mathbf{v}'_k - 2 \frac{\mathbf{v}' \cdot \nabla_{\mathbf{x}} U_M}{(\nabla_{\mathbf{x}} U_M)^T \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M} (\mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M)_k \right) \right] \right. \\
 &\quad \left. + \max \left[0, -\nabla_{\mathbf{x}_k} U_M \cdot \mathbf{v}'_k \right] \right\} \\
 &= \sum_{k \in \mathcal{N}} \left\{ \max \left[0, \nabla_{\mathbf{x}_k} U_M \cdot \left(\mathbf{v}'_k - 2 \frac{\mathbf{v}' \cdot \nabla_{\mathbf{x}} U_M}{(\nabla_{\mathbf{x}} U_M)^T \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M} (\mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M)_k \right) \right] \right. \\
 &\quad \left. + \max \left[0, \nabla_{\mathbf{x}_k} U_M \cdot \mathbf{v}'_k \right] \right\}.
 \end{aligned} \tag{10}$$

The second equation here follows from expressing the equality

$$\mathbf{v} \cdot \nabla_{\mathbf{x}} U_M + \left(\mathbf{v} - 2 \frac{\mathbf{v} \cdot \nabla_{\mathbf{x}} U_M}{(\nabla_{\mathbf{x}} U_M)^T \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M} \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M \right) \cdot \nabla_{\mathbf{x}} U_M = 0 \tag{11}$$

as a sum over $k \in \mathcal{N}$ and from grouping the positive and negative terms separately. The normalization factor C in $p_{(\mathbf{v}',j),(\mathbf{v},i)}^M$ is the same for every possible next active atom i . Because of the $\delta^{(3N_a)}$ -functions in Eq. (8), the normalization factor C can also be written in terms of the velocity \mathbf{v} after the force kick by replacing $\mathbf{v}' \rightarrow \mathbf{v}$ in Eq. (10). This yields

$$\begin{aligned}
 \mathcal{F}_M^{\text{lift}}(\mathbf{x}, \mathbf{v}, i) &= \beta \hat{\pi}(\mathbf{x}, \mathbf{v}, i) \max \left[0, -\nabla_{\mathbf{x}_i} U_M \cdot \mathbf{v}_i \right] \frac{1}{C} \sum_{j \in \mathcal{N}} \left\{ \right. \\
 &\quad \max \left[0, \nabla_{\mathbf{x}_j} U_M \cdot \left(\mathbf{v}_j - 2 \frac{\mathbf{v} \cdot \nabla_{\mathbf{x}} U_M}{(\nabla_{\mathbf{x}} U_M)^T \mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M} (\mathbf{M}^{-1} \nabla_{\mathbf{x}} U_M)_j \right) \right] \\
 &\quad \left. + \max \left[0, \nabla_{\mathbf{x}_j} U_M \cdot \mathbf{v}_j \right] \right\} \\
 &= \beta \hat{\pi}(\mathbf{x}, \mathbf{v}, i) \max \left[0, -\nabla_{\mathbf{x}_i} U_M \cdot \mathbf{v}_i \right] \\
 &= -\mathcal{F}_M^{\text{phys}}(\mathbf{x}, \mathbf{v}, i),
 \end{aligned} \tag{12}$$

where we used that the velocity \mathbf{v} and activity label i are uniformly distributed over their sample spaces $\mathcal{V}^{3N_a}(E_{\text{kin}})$ and \mathcal{N} , respectively. Equation (12) concludes the proof that the presented Newtonian-general scheme for the update of the lifting variables in an event by a general factor potential satisfies the global-balance condition of Eq. (5). The reflective²⁹ and forward⁴⁰ event-chain Monte Carlo variants can likewise be generalized to factor potentials depending on an arbitrary number of atoms.

For the Newtonian-pair scheme for distance-dependent pair potentials and identity mass matrix $\mathbf{M} = \mathbf{I}$, we get

$$\begin{aligned}
 p_{(\mathbf{v}',j),(\mathbf{v},i)}^M &= \begin{cases} \delta^{(3N_a)} \left(\mathbf{v} - \mathbf{v}' + 2 \frac{\mathbf{v}' \cdot \nabla_{\mathbf{x}} U_M}{|\nabla_{\mathbf{x}} U_M|^2} \nabla_{\mathbf{x}} U_M \right) & \text{if } i, j \in M \text{ and } i \neq j, \\ 0 & \text{otherwise,} \end{cases} \\
 &= \begin{cases} \delta^{(3N_a)} \left(\mathbf{v}' - \mathbf{v} + 2 \frac{\mathbf{v}' \cdot \nabla_{\mathbf{x}} U_M}{|\nabla_{\mathbf{x}} U_M|^2} \nabla_{\mathbf{x}} U_M \right) & \text{if } i, j \in M \text{ and } i \neq j, \\ 0 & \text{otherwise.} \end{cases}
 \end{aligned} \tag{13}$$

With $i, j \in M$ and $i \neq j$, this yields

$$\begin{aligned}
 \mathcal{F}_M^{\text{lift}}(\mathbf{x}, \mathbf{v}, i) &= \beta \hat{\pi}(\mathbf{x}, \mathbf{v}, i) \max \left[0, \nabla_{\mathbf{x}_j} U_M \cdot \left(\mathbf{v}_j - 2 \frac{\mathbf{v} \cdot \nabla_{\mathbf{x}} U_M}{|\nabla_{\mathbf{x}} U_M|^2} \nabla_{\mathbf{x}_j} U_M \right) \right] \\
 &= \beta \hat{\pi}(\mathbf{x}, \mathbf{v}, i) \max \left[0, \nabla_{\mathbf{x}_j} U_M \cdot \mathbf{v}_j - 2 \frac{\mathbf{v}_i \cdot \nabla_{\mathbf{x}_i} U_M + \mathbf{v}_j \cdot \nabla_{\mathbf{x}_j} U_M}{|\nabla_{\mathbf{x}_i} U_M|^2 + |\nabla_{\mathbf{x}_j} U_M|^2} |\nabla_{\mathbf{x}_j} U_M|^2 \right] \\
 &= \beta \hat{\pi}(\mathbf{x}, \mathbf{v}, i) \max \left[0, -\nabla_{\mathbf{x}_i} U_M \cdot \mathbf{v}_i \right] \\
 &= -\mathcal{F}_M^{\text{phys}}(\mathbf{x}, \mathbf{v}, i).
 \end{aligned} \tag{14}$$

Here, we again used the uniform distributions of \mathbf{v} and i and the translational invariance $\nabla_{\mathbf{x}_i} U_M = -\nabla_{\mathbf{x}_j} U_M$. Thus, also the Newtonian-pair scheme satisfies the global-balance condition of Eq. (5).

Fibonacci vectors

The cell-veto algorithm was previously only used for the straight variant of event-chain Monte Carlo^{23,24,27}. Its finite set of possible velocities \mathcal{D} of the single active atom a allows one to calculate a Walker table for every velocity $\mathbf{v}_a \in \mathcal{D}$. For instance, for factor pair potentials between the active atom a and another atom b at positions \mathbf{x}_a and \mathbf{x}_b , $U_M(\mathbf{x}) = U_M(\mathbf{x}_a, \mathbf{x}_b)$, every cell bound $q_M^{\text{cell}}(C_a, C_b, \mathbf{v}_a)$ for the pair of cells C_a and C_b in the Walker table may be written as

$$q_M^{\text{cell}}(C_a, C_b, \mathbf{v}_a) = \max_{\mathbf{x}_a \in C_a, \mathbf{x}_b \in C_b} \beta \max [0, \nabla_{\mathbf{x}_a} U_M(\mathbf{x}_a, \mathbf{x}_b) \cdot \mathbf{v}_a]. \quad (15)$$

In principle, a Walker table must be precomputed for every velocity $\mathbf{v}_a \in \mathcal{D}$ and for every possible cell C_a of the active atom. Symmetries, such as a translational invariance of the factor potential, may heavily reduce the necessary number of Walker tables²³. During an actual simulation with the straight event-chain Monte Carlo algorithm, the relevant Walker table is determined by the cell of the currently active atom and its velocity. The generalization of Eq. (15) to more complex factors is straightforward. For example, in order to treat the Coulomb interaction between two water molecules in the SPC/Fw water model, the position and orientation of the water molecule containing the active atom may be varied in C_a , while the position and orientation of the other molecule is varied in C_b . From a practical point of view, the cell bounds $q_M^{\text{cell}}(C_a, C_b)$ are usually not computed exactly but rather approximated, e.g., by considering a finite set of positions $\mathbf{x}_a \in C_a$ and $\mathbf{x}_b \in C_b$, and including a corrective multiplicative prefactor²⁴. As long as the approximated cell bound satisfies $\tilde{q}_M^{\text{cell}}(C_a, C_b, \mathbf{v}_a) \geq q_M^{\text{cell}}(C_a, C_b, \mathbf{v}_a)$, a Poisson thinning procedure corrects *any* overestimate²⁶. The quality of $\tilde{q}_M^{\text{cell}}(C_a, C_b, \mathbf{v}_a)$, however, does influence the performance. Higher cell bounds in the cell-veto algorithm yield more events per unit distance that have to be confirmed by computing the actual event rate.

The inherent discretization of the continuous-position space in Eq. (15) can be translated to a continuous velocity space, as it appears, e.g., in generalized Newtonian event-chain Monte Carlo. Consider a finite number of unit vectors $\hat{\mathbf{v}}_d$ on the two-dimensional unit sphere, and let \mathcal{V}_d be the associated Voronoi cells under some distance function. We can then formally compute the cell bounds for every Voronoi cell as

$$q_M^{\text{cell}}(C_a, C_b, \mathcal{V}_d) = \max_{\mathbf{x}_a \in C_a, \mathbf{x}_b \in C_b, \mathbf{v}_a \in \mathcal{V}_d} \beta \max [0, \nabla_{\mathbf{x}_a} U_M(\mathbf{x}_a, \mathbf{x}_b) \cdot \mathbf{v}_a]. \quad (16)$$

The results are used to precompute Walker tables for every Voronoi cell \mathcal{V}_d and for every cell C_a , where symmetries may again heavily reduce the number of actually necessary tables. During the generalized Newtonian event-chain Monte Carlo simulation, the relevant Walker table is determined by finding the cell of the currently active atom and the Voronoi cell of its normalized velocity $\mathbf{v}_a/|\mathbf{v}_a|$. We can then correct all cell bounds in the relevant Walker table to the actual speed of the active atom by multiplying them with $|\mathbf{v}_a|$.

In this paper, as proof of concept, we map a generalized Fibonacci lattice onto a two-dimensional unit sphere by the Lambert cylindrical equal-area projection to generate D unit vectors $\hat{\mathbf{v}}_d^{\text{fib}}$:

$$\hat{\mathbf{v}}_d^{\text{fib}} = (\cos \phi_d \sin \theta_d, \sin \phi_d \sin \theta_d, \cos \theta_d)^T, \quad (17)$$

where $0 \leq d < D$ and

$$\begin{aligned} \phi_d &= \frac{2\pi d}{\varphi}, \\ \theta_d &= \arccos \left(1 - \frac{2(d + \varepsilon)}{D - 1 + 2\varepsilon} \right), \end{aligned} \quad (18)$$

with the golden ratio $\varphi = (1 + \sqrt{5})/2$ ^{63–65}. The empirical choice of the parameter $\varepsilon = 0.36$ optimizes the average nearest-neighbor distance of the Fibonacci vectors $\hat{\mathbf{v}}_d$. As a distance function to construct the Voronoi cells \mathcal{V}_d , we use the quick-to-evaluate cosine distance. Equations (17) and (18) efficiently generate an arbitrary number of Fibonacci vectors with a nearly uniform distribution. However, there is no efficient inverse mapping from a general velocity \mathbf{v}_a to the closest Fibonacci vector $\hat{\mathbf{v}}_d$. This is not a problem in this paper because we choose $D = 10$ small. Then, $\hat{\mathbf{v}}_d$ can be found by brute force. If considerably larger values of D become necessary, other point configurations on the two-dimensional unit sphere may be considered^{65,66}. For the uniform SPC/Fw water systems in this paper, the Walker tables do not strongly depend on the respective Fibonacci vector. The proper discretization of velocity space will, however, impact the performance of Newtonian event-chain Monte Carlo in nonuniform systems.

SPC/Fw water model

This paper benchmarks the event-chain Monte Carlo algorithm, molecular dynamics, and the Metropolis algorithm for molecular simulations of the atomistic flexible simple point-charge SPC/FW water model³⁴. It defines a water molecule by three charged interaction sites that represent the oxygen and hydrogen atoms. We treat the canonical ensemble in a cubic box with periodic boundary conditions, i.e., N water molecules with $N_a = 3N$ atoms in a periodically repeated cubic box of side length L at a given temperature $T \sim 300$ K. Within any water molecule $i \in \{1, \dots, N\}$, the intramolecular interactions of the SPC/Fw model consist of two harmonic bond potentials $U_{\text{bond}}^{i,1}$ and $U_{\text{bond}}^{i,2}$ that lead to fluctuations of the O–H bond lengths around their equilibrium length. Likewise, a harmonic bending potential U_{bend}^i yields a fluctuation of the H–O–H opening angle around an equilibrium value. The intermolecular interactions between two different water molecules i and j consist of a Lennard-Jones potential U_L^{ij} between the two oxygen atoms, and a Coulomb potential U_C^{ij} between all nine pairs of charged atoms. The intermolecular potentials explicitly include the interactions between all periodic images of the two involved water molecules. The total potential U of an all-atom configuration \mathbf{x} is a sum of these factor potentials:

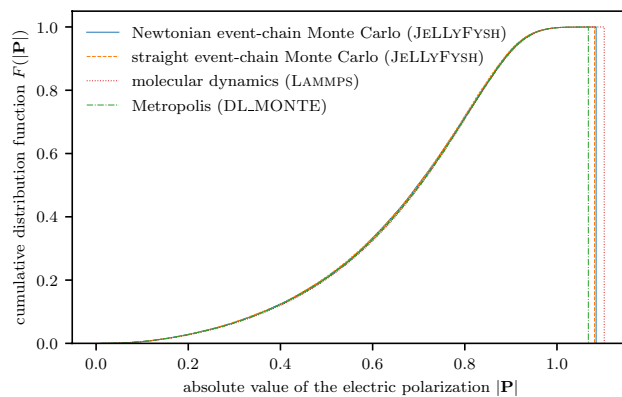


Figure 5. Validation for the SPC/Fw water model. Cumulative distribution function of the absolute value of the electric polarization of $N = 2$ SPC/Fw water molecules in a cubic box of side length $L = 20 \text{ \AA}$.

$$U(\mathbf{x}) = \sum_{i=1}^N \left[U_{\text{bond}}^{i,1}(\mathbf{x}) + U_{\text{bond}}^{i,2}(\mathbf{x}) + U_{\text{bend}}^i(\mathbf{x}) \right] + \sum_{i=1}^N \sum_{j=1}^{i-1} \left[U_{\text{LJ}}^{ij}(\mathbf{x}) + U_{\text{C}}^{ij}(\mathbf{x}) \right]. \quad (19)$$

The bond and Lennard-Jones factor potentials depend on two atomic positions; the bending and Coulomb factor potentials depend on three and six atomic positions, respectively (see also Ref.²³, Section V A). The empirical parameters of the different potentials of the SPC/Fw water model are as in Ref.³⁴.

Simulation protocols—molecular dynamics

We use the feature release from February 8, 2023, of LAMMPS^{8,67} for the molecular-dynamics simulations of the SPC/Fw water model in this paper. We employ a spherical cut-off for the Lennard-Jones potential. The Coulomb potential is treated by a particle–particle particle–mesh solver¹². Unless explicitly specified otherwise (as, e.g., in Fig. 4b), the solver uses a target accuracy of 10^{-6} . Coulomb interactions are partly treated in discretized reciprocal space, whereby the grid size is chosen to meet the target accuracy based on analytic error estimates obtained from a specific global charge distribution^{15,48,49}. Thermostatting is achieved by integrating the Nosé–Hoover-chain equations of motion for the canonical ensemble⁶⁸ with a time-reversible measure-preserving Verlet integrator⁶⁹. The time step is 1 fs, and the temperature is relaxed in a timespan of roughly 300 time steps. This simulation protocol for molecular dynamics in LAMMPS is validated by comparing numerical results for $N = 2$ SPC/Fw water molecules of all sampling methods of this paper (see Fig. 5).

Simulation protocols—Metropolis algorithm

We use version 2.07 of the DL_MONTE software package^{47,70,71} to sample the canonical ensemble of the SPC/Fw water model with the reversible Metropolis algorithm³⁷, and minimally amended the software to output the electric polarization. We employ a spherical cut-off for the Lennard-Jones potential. The Coulomb interaction is treated by an Ewald summation⁷² with a target-accuracy tolerance of 10^{-6} . The simulations with the non-reversible event-chain Monte Carlo algorithm in this paper move a single atom at a time. For a direct comparison to the dynamics of the reversible Metropolis algorithm (see Fig. 3c), each trial proposes a new position of a single random atom in a proposal cube around its original position. The new position is accepted with the usual Metropolis criterion based on the change of energy. The size of the proposal cube is separately adapted for the hydrogen and oxygen atoms during the simulation to obtain a target acceptance rate of 37%. This simulation protocol for the Metropolis algorithm in DL_MONTE is validated by comparing numerical results for $N = 2$ SPC/Fw water molecules of all sampling methods of this paper (see Fig. 5).

Simulation protocols—event-chain Monte Carlo

We develop version 2.0 of JELLYFYSH^{24,56} for the event-chain Monte Carlo simulations of the SPC/Fw water model in this paper. In addition to the straight variant implemented in versions < 2.0 , version 2.0 implements the generalized Newtonian event-chain Monte Carlo variant. We use the factorization in Eq. (19). In particular, we group all the charge–charge Coulomb interactions between two water molecules into a single factor. Theory then predicts an optimal logarithmic increase in the number of events from these molecular Coulomb factors²³ that is numerically confirmed in Fig. 3b.

Straight event-chain Monte Carlo (used in Fig. 3c) periodically aligns the unit velocity $|\mathbf{v}_a| = 1$ of the active atom with the coordinate axes. We change the velocity after a chain time of $\tau_{\text{chain}} = 0.2N^{43}$. In an event triggered by a factor M , the velocity is transferred to another atom which belongs to M . For factors with more than two atoms, the next active atom is chosen according to the ratio lifting scheme^{23,39}.

Generalized Newtonian event-chain Monte Carlo assigns a velocity label to every atom, but only moves a single one at any time. The velocities are initialized so that the average speed of the active atom during the simulation is approximately one: $\langle |\mathbf{v}_a| \rangle \approx 1$. This is possible because the velocities have no kinematic meaning as they have in molecular dynamics. The physical temperature only rescales the event rates [see Eq. (1)]. The velocities

are resampled in large time intervals of $\tau_{\text{chain}} = 10000N$. This is because Newtonian event-chain Monte Carlo rotates dipoles most efficiently in the limit $\tau_{\text{chain}} \rightarrow \infty$, while a finite value of τ_{chain} ensures irreducibility^{41,57}. Although our generalized Newtonian event-chain Monte Carlo can consider a general mass matrix of the atoms in its events, we set the masses of all atoms to be equal. Then, the events of factors M of two atoms can be treated with the deterministic Newtonian-pair scheme in which the two involved velocities and the active atom always change. For factors M of more than two atoms, we apply the stochastic Newtonian-general scheme that corresponds to a generalized ratio lifting scheme.

The events of the bond factors are computed exactly [see Eq. (2)], while the bending factors are treated with a piecewise-linear bounding potential followed by a Poisson thinning procedure (see Ref.²⁴, Section 4.4.5). The cell-veto algorithm²⁷ bundles most molecular Coulomb factors and relies on a cell-occupancy system that tracks the molecular barycenters (we generally adapt the setup described in Ref.²⁴, Section 5.3.4). Very close molecule pairs or surplus molecules in the cell-occupancy system are separately treated with a piecewise-linear bounding potential. The cell bounds for 10 different Fibonacci vectors are estimated by varying the position of an atom in one cell, and the position of a dipole in the other cell. Here, the dipole is aligned with the direction of the gradient of the charge–dipole Coulomb interaction. The maximum event rate multiplied by an empirical prefactor yields the cell bound. During the simulation, the cell bounds in the Walker table are rescaled to the actual speed and charge of the active atom. Events from the cell-veto algorithm are confirmed with a Poisson thinning procedure that compares the actual event rate of the Coulomb interaction between two molecules with the sampled cell bound. To compute the real event rate, we compute the gradients of the Coulomb potential between the nine pairs of charged atoms [see also Eq. (1)] with an Ewald summation⁷² that is tuned to machine precision without any assumption on the global charge distribution. We also use the cell-veto algorithm to treat most Lennard-Jones factors, although we cut off the interaction beyond the closest images for consistency with the other sampling algorithms. We carefully checked that an alternative spherical cut-off does not change the results of this paper. The cell-veto algorithm for the Lennard-Jones interaction relies on a cell-occupancy system that tracks oxygen atoms. Very close or surplus oxygen pairs are treated directly [see Eq. (2)]. The cell bounds for 10 different Fibonacci vectors are estimated by varying the oxygen positions evenly in the two cells, and again including an empirical prefactor. The parameters of the cell-occupancy systems, as the cell sizes and the number of nearby cells that are excluded from the cell-veto algorithm, are tuned to minimize computer time. As a rule of thumb, the largest possible cells that do not yield any surplus water molecules, and two excluded layers are found to be a good choice.

These simulation protocols for straight and generalized Newtonian event-chain Monte Carlo in JELLYFYSH are validated by comparing numerical results for $N = 2$ SPC/Fw water molecules of all sampling methods of this paper (see Fig. 5).

Simulation protocols—creation of initial configurations

We create initial configurations in the range from $N = 64$ to $N = 2744$ SPC/Fw water molecules with hydrogen mass $m_H = 1.0079$ Da and oxygen mass $m_O = 15.9994001$ Da at a density of $\rho = 0.97$ g cm⁻³ using the software package PLAYMOL⁷³ (commit 67eb56c from 26 November 2019). Initial configurations are equilibrated using a molecular-dynamics simulation of LAMMPS in the isothermal–isobaric ensemble at pressure $P = 1$ atm. Similar to thermostatting, barostatting is achieved by integrating the appropriate Nosé–Hoover-chain equations of motion with a Verlet integrator^{68,69} with a time step of 1 fs. This results in initial configurations for the simulations in the canonical ensemble with the different sampling algorithms at slightly different densities for different N . We confirmed that these small density variations do not influence the results of this paper.

Analysis of electric polarization

We sample the electric polarization \mathbf{P} (or the total electric dipole moment $\mathbf{P} = \sum_{i=1}^N \mathbf{p}_i$, where \mathbf{p}_i is the molecular dipole moment of the water molecule i) of the SPC/Fw water system with the different sampling algorithms. The generated “time”-series $\mathbf{P}(t)$ (where the “time” t only has a physical meaning in molecular dynamics) yields the normalized autocorrelation function

$$\Phi_{\mathbf{P}}(\tau) = \frac{\langle \mathbf{P}(t) \cdot \mathbf{P}(t + \tau) \rangle_t}{\langle \mathbf{P}(t) \cdot \mathbf{P}(t) \rangle_t}, \quad (20)$$

where we explicitly assume that $\langle \mathbf{P} \rangle = 0$ for $t \rightarrow \infty$. The first part of each trajectory in the canonical ensemble is not considered in the equilibrium average. We observe an exponential decay in all sampling algorithms and fit an exponential $\sim \exp(-\tau/\tau_{\Phi})$ to extract the time constant τ_{Φ} and its standard error (see also Ref.³⁴). We naïvely parallelize the Metropolis and event-chain Monte Carlo algorithms by running 20 to 50 independent simulations (where the number of runs grows with N). For every τ , we then compute the median autocorrelation function $\Phi_{\mathbf{P}}(\tau)$ and estimate its error with the bootstrap method. These errors are then considered in the least-squares fit.

The units of the autocorrelation time τ_{Φ} for the different sampling algorithms are physical time for molecular dynamics, Monte-Carlo trials for the Metropolis algorithm, and continuous Monte-Carlo time for the event-chain Monte Carlo algorithms. As a first measure of the efficiency of the different dynamics, we compare the autocorrelation distances d_{Φ} (see Fig. 3c). It gives the average cumulative distance moved by the atoms in each of the sampling schemes. Since only a single atom moves in our implementation of the event-chain Monte Carlo algorithm, it follows that $d_{\Phi}^{\text{ECMC}} = \langle |\mathbf{v}_a| \rangle^{\text{ECMC}} \tau_{\Phi}^{\text{ECMC}}$, where $\langle |\mathbf{v}_a| \rangle^{\text{ECMC}}$ is the average speed of the active atom during the simulation. For molecular dynamics, we can compute the average speed of the hydrogen and oxygen atoms $t \in \{H, O\}$ with the Maxwell–Boltzmann distribution:

$$\langle |\mathbf{v}_t| \rangle^{\text{MD}} = \sqrt{\frac{8k_{\text{B}}T}{\pi m_t}}, \quad (21)$$

which yields $d_{\Phi}^{\text{MD}} = N(2\langle |\mathbf{v}_H| \rangle^{\text{MD}} + \langle |\mathbf{v}_O| \rangle^{\text{MD}}) \tau_{\Phi}^{\text{MD}}$. In the Metropolis algorithm, each trial samples a random displacement vector $\mathbf{d}_t = [\text{ran}(-\delta_t, \delta_t), \text{ran}(-\delta_t, \delta_t), \text{ran}(-\delta_t, \delta_t)]^T$ for a random atom, where δ_t is tuned independently for the oxygen and hydrogen atoms $t \in \{H, O\}$ to accept the displacement with probability $p \approx 0.37$. We can compute the average length $\langle |\mathbf{d}_t| \rangle^{\text{Met}}$ of the sampled displacement vector as

$$\langle |\mathbf{d}_t| \rangle^{\text{Met}} = \frac{1}{8\delta_t^3} \int_{-\delta_t}^{\delta_t} dx \int_{-\delta_t}^{\delta_t} dy \int_{-\delta_t}^{\delta_t} dz \sqrt{x^2 + y^2 + z^2} \approx 0.961 \delta_t. \quad (22)$$

Neglecting correlations between rejection probability and the displacement for this first measure of efficiency, we get $d_{\Phi}^{\text{Met}} = p(2\langle |\mathbf{d}_H| \rangle^{\text{Met}}/3 + \langle |\mathbf{d}_O| \rangle^{\text{Met}}/3) \tau_{\Phi}^{\text{Met}}$.

The computer autocorrelation time of our generalized Newtonian event-chain Monte Carlo algorithm (see Fig. 4b) is obtained by combining the autocorrelation distance (see Fig. 3c), the event rates (see Fig. 3b), and the computer time per event in JELLYFYSH (see Fig. 3a). Similarly, we measure the computer time per time step of molecular dynamics in LAMMPS for different target accuracies of its particle–particle time–mesh solver (see Fig. 4a). This is then combined with the autocorrelation time in the number of time steps.

Code availability

The JELLYFYSH software is made available under the GNU GPLv3 license at <https://github.com/jellyfysh>. All configuration files for the simulations of this paper are part of JELLYFYSH.

Received: 5 December 2023; Accepted: 27 June 2024

Published online: 16 July 2024

References

- Feynman, R., Leighton, R. & Sands, M. *The Feynman Lectures on Physics, Vol. I: Mainly Mechanics, Radiation, and Heat* Vol. 1 of *The Feynman Lectures on Physics* (Addison-Wesley Publishing Company, 1963).
- Shaw, D. E. *et al.* Atomic-level characterization of the structural dynamics of proteins. *Science* **330**, 341–346. <https://doi.org/10.1126/science.1187409> (2010).
- Schlick, T. *Molecular Modeling and Simulation: An Interdisciplinary Guide* (Springer, 2002).
- Karplus, M. Development of multiscale models for complex chemical systems: From H+H₂ to biomolecules (Nobel lecture). *Angew. Chem. Int. Ed.* **53**, 9992–10005. <https://doi.org/10.1002/anie.201403924> (2014).
- Case, D. A. *et al.* The Amber biomolecular simulation programs. *J. Comput. Chem.* **26**, 1668–1688. <https://doi.org/10.1002/jcc.20290> (2005).
- Brooks, B. R. *et al.* CHARMM: The biomolecular simulation program. *J. Comput. Chem.* **30**, 1545–1614. <https://doi.org/10.1002/jcc.21287> (2009).
- Berendsen, H. J. C., van der Spoel, D. & van Drunen, R. GROMACS: A message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.* **91**, 43–56. [https://doi.org/10.1016/0010-4655\(95\)00042-E](https://doi.org/10.1016/0010-4655(95)00042-E) (1995).
- Thompson, A. P. *et al.* LAMMPS—a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.* **271**, 108171. <https://doi.org/10.1016/j.cpc.2021.108171> (2022).
- Phillips, J. C. *et al.* Scalable molecular dynamics on CPU and GPU architectures with NAMD. *J. Chem. Phys.* **153**, 044130. <https://doi.org/10.1063/5.0014475> (2020).
- Hammonds, K. D. & Heyes, D. M. Shadow Hamiltonian in classical NVE molecular dynamics simulations: A path to long time stability. *J. Chem. Phys.* **152**, 024114. <https://doi.org/10.1063/1.5139708> (2020).
- Wong-ekkabut, J. & Karttunen, M. The good, the bad and the user in soft matter simulations. *Biochim. Biophys. Acta Biomembr.* **1858**, 2529–2538. <https://doi.org/10.1016/j.bbamem.2016.02.004> (2016).
- Hockney, R. W. & Eastwood, J. W. *Computer Simulation Using Particles* (CRC Press, 1988).
- Darden, T., York, D. & Pedersen, L. Particle mesh Ewald: An $N \log(N)$ method for Ewald sums in large systems. *J. Chem. Phys.* **98**, 10089–10092. <https://doi.org/10.1063/1.464397> (1993).
- Essmann, U. *et al.* A smooth particle mesh Ewald method. *J. Chem. Phys.* **103**, 8577–8593. <https://doi.org/10.1063/1.470117> (1995).
- Petersen, H. G. Accuracy and efficiency of the particle mesh Ewald method. *J. Chem. Phys.* **103**, 3668–3679. <https://doi.org/10.1063/1.470043> (1995).
- Kohnke, B., Kutzner, C. & Grubmüller, H. A GPU-accelerated fast multipole method for GROMACS: Performance and accuracy. *J. Chem. Theory Comput.* **16**, 6938–6949. <https://doi.org/10.1021/acs.jctc.0c00744> (2020).
- Kittel, C. *Introduction to Solid State Physics* 8th edn. (Wiley, 2005).
- Smit, B. & Frenkel, D. Vapor–liquid equilibria of the two-dimensional Lennard-Jones fluid(s). *J. Chem. Phys.* **94**, 5663–5668. <https://doi.org/10.1063/1.460477> (1991).
- Smit, B. Phase diagrams of Lennard-Jones fluids. *J. Chem. Phys.* **96**, 8639–8640. <https://doi.org/10.1063/1.462271> (1992).
- Lei, Y. A., Bykov, T., Yoo, S. & Zeng, X. C. The Tolman length: Is it positive or negative?. *J. Am. Chem. Soc.* **127**, 15346–15347. <https://doi.org/10.1021/ja054297i> (2005).
- Tröster, A., Oettel, M., Block, B., Virnau, P. & Binder, K. Numerical approaches to determine the interface tension of curved interfaces from free energy calculations. *J. Chem. Phys.* **136**, 064709. <https://doi.org/10.1063/1.3685221> (2012).
- Wennberg, C. L., Murtola, T., Hess, B. & Lindahl, E. Lennard-Jones lattice summation in bilayer simulations has critical effects on surface tension and lipid properties. *J. Chem. Theory Comput.* **9**, 3527–3537. <https://doi.org/10.1021/ct400140n> (2013).
- Faulkner, M. F., Qin, L., Maggs, A. C. & Krauth, W. All-atom computations with irreversible Markov chains. *J. Chem. Phys.* **149**, 064113. <https://doi.org/10.1063/1.5036638> (2018).
- Höllmer, P., Qin, L., Faulkner, M. F., Maggs, A. C. & Krauth, W. JeLLyFysh-version10 – a Python application for all-atom event-chain Monte Carlo. *Comput. Phys. Commun.* **253**, 107168. <https://doi.org/10.1016/j.cpc.2020.107168> (2020).
- Michel, M., Kapfer, S. C. & Krauth, W. Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps. *J. Chem. Phys.* **140**, 054116. <https://doi.org/10.1063/1.4863991> (2014).
- Lewis, P. A. W. & Shedler, G. S. Simulation of nonhomogeneous Poisson processes by thinning. *Nav. Res. Logist. Q.* **26**, 403–413. <https://doi.org/10.1002/nav.3800260304> (1979).

27. Kapfer, S. C. & Krauth, W. Cell-veto Monte Carlo algorithm for long-range systems. *Phys. Rev. E* **94**, 031302. <https://doi.org/10.1103/PhysRevE.94.031302> (2016).
28. Davis, M. H. A. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **46**, 353–376. <https://doi.org/10.1111/j.2517-6161.1984.tb01308.x> (1984).
29. Bernard, E. P., Krauth, W. & Wilson, D. B. Event-chain Monte Carlo algorithms for hard-sphere systems. *Phys. Rev. E* **80**, 056704. <https://doi.org/10.1103/PhysRevE.80.056704> (2009).
30. Klement, M. & Engel, M. Efficient equilibration of hard spheres with Newtonian event chains. *J. Chem. Phys.* **150**, 174108. <https://doi.org/10.1063/1.5090882> (2019).
31. Walker, A. J. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.* **3**, 253–256. <https://doi.org/10.1145/355744.355749> (1977).
32. Tartero, G. & Krauth, W. Fast sampling for particle systems with long-range potentials. [arXiv:2406.00184](https://arxiv.org/abs/2406.00184).
33. Bernard, E. P. & Krauth, W. Two-step melting in two dimensions: First-order liquid-hexatic transition. *Phys. Rev. Lett.* **107**, 155704. <https://doi.org/10.1103/PhysRevLett.107.155704> (2011).
34. Wu, Y., Tepper, H. L. & Voth, G. A. Flexible simple point-charge water model with improved liquid-state properties. *J. Chem. Phys.* **124**, 024503. <https://doi.org/10.1063/1.2136877> (2006).
35. de Leeuw, S. W., Perram, J. W. & Smith, E. R. Simulation of electrostatic systems in periodic boundary conditions. I. Lattice sums and dielectric constants. *Proc. R. Soc. A* **373**, 27–56. <https://doi.org/10.1098/rspa.1980.0135> (1980).
36. Krauth, W. Event-chain Monte Carlo: Foundations, applications, and prospects. *Front. Phys.* **9**, 663457663457. <https://doi.org/10.3389/fphy.2021.663457> (2021).
37. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092. <https://doi.org/10.1063/1.1699114> (1953).
38. Gamma, E., Helm, R., Johnson, R. & Vlissides, J. M. *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley Professional, 1994).
39. Harland, J., Michel, M., Kampmann, T. A. & Kierfeld, J. Event-chain Monte Carlo algorithms for three- and many-particle interactions. *Europhys. Lett.* **117**, 30001. <https://doi.org/10.1209/0295-5075/117/30001> (2017).
40. Michel, M., Durmus, A. & Sénécal, S. Forward event-Chain Monte Carlo: Fast sampling by randomness control in irreversible Markov chains. *J. Comput. Graph. Stat.* **29**, 689–702. <https://doi.org/10.1080/10618600.2020.1750417> (2020).
41. Höllmer, P., Maggs, A. C. & Krauth, W. Hard-disk dipoles and non-reversible Markov chains. *J. Chem. Phys.* **156**, 084108. <https://doi.org/10.1063/5.0080101> (2022).
42. Li, B., Todo, S., Maggs, A. C. & Krauth, W. Multithreaded event-chain Monte Carlo with local times. *Comput. Phys. Commun.* **261**, 107702. <https://doi.org/10.1016/j.cpc.2020.107702> (2021).
43. Qin, L. Application of irreversible Monte Carlo in realistic long-range systems. Phd thesis, Université Paris Sciences et Lettres (2020). <https://tel.archives-ouvertes.fr/tel-02998657>.
44. Tartero, G. & Krauth, W. Concepts in Monte Carlo sampling. *Am. J. Phys.* **92**, 65–77. <https://doi.org/10.1119/5.0176853> (2024).
45. Saunders, W. R., Grant, J. & Müller, E. H. A new algorithm for electrostatic interactions in Monte Carlo simulations of charged particles. *J. Comput. Phys.* **430**, 110099. <https://doi.org/10.1016/j.jcp.2020.110099> (2021).
46. Greengard, L. & Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **73**, 325–348. [https://doi.org/10.1016/0021-9991\(87\)90140-9](https://doi.org/10.1016/0021-9991(87)90140-9) (1987).
47. Purton, J. A., Crabtree, J. C. & Parker, S. C. DL_MONTE: A general purpose program for parallel Monte Carlo simulation. *Mol. Simul.* **39**, 1240–1252. <https://doi.org/10.1080/08927022.2013.839871> (2013).
48. Kolafa, J. & Perram, J. W. Cutoff errors in the Ewald summation formulae for point charge systems. *Mol. Simul.* **9**, 351–368. <https://doi.org/10.1080/08927029208049126> (1992).
49. Deserno, M. & Holm, C. How to mesh up Ewald sums. II. An accurate error estimate for the particle-particle-particle-mesh algorithm. *J. Chem. Phys.* **109**, 7694–7701. <https://doi.org/10.1063/1.477415> (1998).
50. Levin, D. A., Peres, Y. & Wilmer, E. L. Markov Chains and Mixing Times (American Mathematical Society, 2008).
51. Vanetti, P., Bouchard-Côté, A., Deligiannidis, G. & Doucet, A. Piecewise-deterministic Markov Chain Monte Carlo. [arXiv:1707.05296](https://arxiv.org/abs/1707.05296) (2018).
52. Fearnhead, P., Bierkens, J., Pollock, M. & Roberts, G. O. Piecewise deterministic Markov processes for continuous-time Monte Carlo. *Stat. Sci.* **33**, 386–412. <https://doi.org/10.1214/18-STS648> (2018).
53. Bierkens, J., Grazi, S., Kamatani, K. & Roberts, G. Daumé III, H. & Singh, A. (eds) *The boomerang sampler*. (eds Daumé III, H. & Singh, A.) *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, 908–918 (2020). <https://proceedings.mlr.press/v119/bierkens20a.html>.
54. Lei, Z., Krauth, W. & Maggs, A. C. Event-chain Monte Carlo with factor fields. *Phys. Rev. E* **99**, 043301. <https://doi.org/10.1103/PhysRevE.99.043301> (2019).
55. Essler, F. H. L. & Krauth, W. Lifted TASEP: a Bethe ansatz integrable paradigm for non-reversible Markov chains (2023). [arXiv:2306.13059](https://arxiv.org/abs/2306.13059).
56. Höllmer, P., Maggs, A. C. & Krauth, W. JeLLyFysh, version 2.0-alpha (2023). <https://github.com/jellyfysh/JeLLyFysh/tree/Versi%202.0-alpha>. Last accessed 04 Oct 2023.
57. Höllmer, P., Noirault, N., Li, B., Maggs, A. C. & Krauth, W. Sparse hard-disk packings and local Markov chains. *J. Stat. Phys.* **187**, 31. <https://doi.org/10.1007/s10955-022-02908-4> (2022).
58. Monemvassitis, A., Guillin, A. & Michel, M. PDMP characterisation of event-chain Monte Carlo algorithms for particle systems. *J. Stat. Phys.* **190**, 66. <https://doi.org/10.1007/s10955-023-03069-8> (2023).
59. Chen, F., Lovász, L. & Pak, I. Lifting Markov Chains to Speed up Mixing. *Proceedings of the thirty-first annual ACM symposium on Theory of Computing* 275–281 (1999). <https://doi.org/10.1145/301250.301315>.
60. Diaconis, P., Holmes, S. & Neal, R. M. Analysis of a nonreversible Markov chain sampler. *Ann. Appl. Probab.* **10**, 726–752. <https://doi.org/10.1214/aoap/1019487508> (2000).
61. Qin, L., Höllmer, P. & Krauth, W. Direction-sweep Markov chains. *J. Phys. A: Math. Theor.* **55**, 105003. <https://doi.org/10.1088/1751-8121/ac508a> (2022).
62. Peters, E. A. J. F. & de With, G. Rejection-free Monte Carlo sampling for general potentials. *Phys. Rev. E* **85**, 026703. <https://doi.org/10.1103/PhysRevE.85.026703> (2012).
63. Roberts, M. How to Evenly Distribute Points on a Sphere More Effectively than the Canonical Fibonacci Lattice (2020). <https://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/>. Last accessed 04 Oct 2023.
64. Feng, Y. An effective energy-conserving contact modelling strategy for spherical harmonic particles represented by surface triangular meshes with automatic simplification. *Comput. Methods Appl. Mech. Eng.* **379**, 113750. <https://doi.org/10.1016/j.cma.2021.113750> (2021).
65. Hardin, D. P., Michaels, T. J. & Saff, E. B. A Comparison of Popular Point Configurations on S^2 . [arXiv:1607.04590](https://arxiv.org/abs/1607.04590) [math.NA] (2016). <https://doi.org/10.48550/arXiv.1607.04590>.
66. Keinert, B., Innmann, M., Sängler, M. & Stamminger, M. Spherical Fibonacci mapping. *ACM Trans. Graph.* **34**, 193. <https://doi.org/10.1145/2816795.2818131> (2015).

67. Plimpton, S., Thompson, A., Moore, S., Kohlmeyer, A. & Berger, R. LAMMPS, Feature release 8 February 2023 (2023). <https://www.lammps.org>. Last accessed 04 Oct 2023.
68. Shinoda, W., Shiga, M. & Mikami, M. Rapid estimation of elastic constants by molecular dynamics simulation under constant stress. *Phys. Rev. B* **69**, 134103. <https://doi.org/10.1103/PhysRevB.69.134103> (2004).
69. Tuckerman, M. E., Alejandre, J., López-Rendón, R., Jochim, A. L. & Martyna, G. J. A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal-isobaric ensemble. *J. Phys. A: Math. Gen.* **39**, 5629. <https://doi.org/10.1088/0305-4470/39/19/S18> (2006).
70. Brukhno, A. V. *et al.* DL_MONTE, version 2.07 (2020). https://gitlab.com/dl_monte/user-hub/-/wikis/home. Last accessed 04 Oct 2023.
71. Brukhno, A. V. *et al.* DL_MONTE: A multipurpose code for Monte Carlo simulation. *Mol. Simul.* **47**, 131–151. <https://doi.org/10.1080/08927022.2019.1569760> (2021).
72. Ewald, P. P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.* **369**, 253–287. <https://doi.org/10.1002/andp.19213690304> (1921).
73. Abreu, C. R. A. Playmol, Commit 67eb56c from 26 November 2019 (2019). <http://atoms.peq.coppe.ufrj.br/playmol/index.html>. Last accessed 04 Oct 2023.

Acknowledgements

We thank L. Qin for helpful discussions. P.H. acknowledges support from the Studienstiftung des deutschen Volkes and from Institut Philippe Meyer. W.K. acknowledges support from the Alexander von Humboldt Foundation.

Author contributions

Conceptualization: P.H., A.C.M., W.K. Formal analysis: P.H. Investigation: P.H., A.C.M. Methodology: P.H., A.C.M., W.K. Project administration: A.C.M., W.K. Software: P.H. Supervision: A.C.M., W.K. Validation: P.H., A.C.M., W.K. Visualization: P.H. Writing—original draft: P.H., W.K. Writing—review & editing: P.H., A.C.M., W.K.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to W.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International

License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024