





RESEARCH ARTICLE | JULY 08 2020

Velocity jump processes: An alternative to multi-timestep methods for faster and accurate molecular dynamics simulations

Special Collection: [Classical Molecular Dynamics \(MD\) Simulations: Codes, Algorithms, Force fields, and Applications](#)

Pierre Monmarché ; Jérémy Weisman; Louis Lagardère ; Jean-Philip Piquemal  



J. Chem. Phys. 153, 024101 (2020)

<https://doi.org/10.1063/5.0005060>



Articles You May Be Interested In

Efficient multiple time step method for use with Ewald and particle mesh Ewald for large biomolecular systems

J. Chem. Phys. (August 2001)

Robust and efficient configurational molecular sampling via Langevin dynamics

J. Chem. Phys. (May 2013)

Extrapolation versus impulse in multiple-timestepping schemes. II. Linear analysis and applications to Newtonian and Langevin dynamics

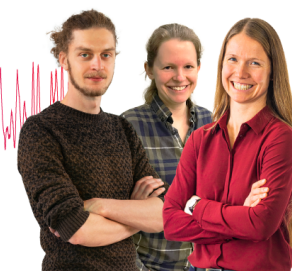
J. Chem. Phys. (August 1998)

Webinar From Noise to Knowledge

May 13th – Register now



Universität
Konstanz



Velocity jump processes: An alternative to multi-timestep methods for faster and accurate molecular dynamics simulations

Cite as: J. Chem. Phys. 153, 024101 (2020); doi: 10.1063/5.0005060

Submitted: 18 February 2020 • Accepted: 15 June 2020 •

Published Online: 8 July 2020



View Online



Export Citation



CrossMark

Pierre Monmarché,^{1,a)}  Jérémy Weisman,² Louis Lagardère,^{2,3,b)}  and Jean-Philip Piquemal^{2,4,c)} 

AFFILIATIONS

¹Sorbonne Université, Laboratoire Jacques-Louis Lions, UMR 7589 CNRS, and Laboratoire de Chimie Théorique, UMR 7616 CNRS, F-75005 Paris, France

²Sorbonne Université, Laboratoire de Chimie Théorique, UMR 7616 CNRS, F-75005 Paris, France

³Institut Parisien de Chimie Physique et Théorique, FR2622 CNRS, F-75005 Paris, France

⁴Institut Universitaire de France, F-75005 Paris, France

Note: This paper is part of the JCP Special Topic on Classical Molecular Dynamics (MD) Simulations: Codes, Algorithms, Force Fields, and Applications.

^{a)} **Electronic mail:** pierre.monmarche@sorbonne-universite.fr

^{b)} **Electronic mail:** louis.lagardere@sorbonne-universite.fr

^{c)} **Author to whom correspondence should be addressed:** jean-philip.piquemal@sorbonne-universite.fr

ABSTRACT

We propose a new route to accelerate molecular dynamics through the use of velocity jump processes allowing for an adaptive time step specific to each atom–atom pair (two-body) interactions. We start by introducing the formalism of the new velocity jump molecular dynamics, ergodic with respect to the canonical measure. We then introduce the new BOUNCE integrator that allows for long-range forces to be evaluated at random and optimal time steps, leading to strong savings in direct space. The accuracy and computational performances of a first BOUNCE implementation dedicated to classical (non-polarizable) force fields are tested in the cases of pure direct-space droplet-like simulations and of periodic boundary conditions (PBC) simulations using Smooth Particle Mesh Ewald method. An analysis of the capability of BOUNCE to reproduce several condensed-phase properties is provided. Since electrostatics and van der Waals two-body contributions are evaluated much less often than with standard integrators using a 1 fs time step, up to a 400% direct-space acceleration is observed. Applying the reversible reference system propagator algorithms [RESPA(1)] to reciprocal-space (many-body) interactions allows BOUNCE-RESPA(1) to maintain large speedups in PBC while maintaining precision. Overall, we show that replacing the BAOAB standard Langevin integrator by the BOUNCE adaptive framework preserves a similar accuracy and leads to significant computational savings.

Published under license by AIP Publishing. <https://doi.org/10.1063/5.0005060>

I. INTRODUCTION

Molecular dynamics (MD) is a popular tool that allows the simulation of complex molecular systems, ranging from materials to biomolecules, mainly used to compute properties by sampling a defined ensemble. In practice, this means that one would like to perform very long trajectories to maximize this sampling. This can be achieved by brute-force high-performance computing

using advanced massively parallel simulation software.^{1–8} However, at the age of exascale computing, any algorithmic enhancements on the statistical physics side would save millions of hours on supercomputers and therefore energy resources. In this connection, the development of more efficient MD integrators is an intense field of research. In recent years, lots of mathematical work have been performed, especially in the framework of the Langevin dynamics, and new techniques emerged such as Leimkuhler's BAOAB^{9,10}

offering stable, accurate, and well-understood integration scheme. Of course, to speed up simulations, one would like primarily to be able to use time steps as large as possible. In practice, multi-timestep approaches¹¹ that are now standard in MD do offer a substantial acceleration through the use of frequency-driven splittings. However, it comes at a price as resonance effects limit the maximum usable time step at a given accuracy.^{12,13} Various alternative strategies have been proposed to enable the use of very large time steps such as in the generalized Langevin equation (GLE)¹⁴ or the stochastic isokinetic extended phase-space algorithm.^{15–17} They are promising but rely either on some empirical fitting¹⁴ or have an important impact on the dynamical properties such as the diffusion coefficient,¹⁷ that is, an indication of a limitation of the sampling rate. In this last case, the computational gain is reduced by the fact that a longer trajectory is necessary to keep the same quality of sampling. To speed up molecular dynamics without resorting to some fitting and while not affecting too much the dynamics,^{14,18} one possibility is to combine well-chosen integrators within a multi-split approach like for BAOAB-RESPA1 to push forward the stability limit.¹⁹

This paper proposes to look outside the box and gives an alternative to multi-timestep approaches. To do so, we will explore the possibility to take into account the slowly varying, bounded part of the atomistic potential at play by a velocity jump mechanism, combining classical MD with probabilistic thinning methods and introducing a new integrator: BOUNCE. The algorithm can somehow be thought of as a multi-timestep integrator, where the long-range forces are evaluated at random time steps. We will first introduce the mathematical framework of the velocity jumps processes and then present the new BOUNCE integrator. Finally, we will evaluate its computational gain and assess its accuracy using classical, non-polarizable force fields, through a set of numerical experiments performed in the framework of a first implementation of BOUNCE within the Tinker-HP software,¹ where various properties will be computed and compared to the state-of-the-art integrators.

II. METHOD

A. Building a new molecular dynamics integrator: Requirements

The main goal of MD is to compute expectations with respect to μ the Boltzmann–Gibbs measure, namely the probability distribution on \mathbb{R}^{6N} with density proportional to $\exp(-\beta H(q, p))$, where $\beta = 1/(k_B T)$ is the inverse temperature, q and p are, respectively, the positions and momenta of N atoms, and the Hamiltonian is $H(q, p) = E_{pot}(q) + E_{kin}(p)$. Denoting M the mass matrix of the system, the kinetic energy is $E_{kin}(p) = \frac{1}{2} p' M^{-1} p$, where x' denotes the transpose of x . The potential energy E_{pot} will be discussed below.

We need to design a (theoretical, at first) trajectory $t \mapsto (q(t), p(t))$ that is ergodic with respect to μ , in the sense that

$$\frac{1}{t} \int_0^t \varphi(q(s), p(s)) ds \xrightarrow[t \rightarrow \infty]{} \int \varphi(q, p) \mu(dq, dp) \quad (1)$$

for all observables φ (in some class of functions on \mathbb{R}^{6N}) and then to approximate the continuous-time dynamics by a numerical scheme

$(q_k, p_k)_{k \in \mathbb{N}}$, so that

$$\frac{1}{K} \sum_{k=1}^K \varphi(q_k, p_k) \xrightarrow[k \rightarrow \infty]{} \int \varphi(q, p) \mu_\delta(dq, dp),$$

where μ_δ is an approximation of μ , depending on the time step δ of the numerical scheme.

A classical process that is ergodic with respect to μ is the Langevin diffusion, solution of the stochastic differential equation (SDE)

$$\begin{cases} \dot{q}(t) = M^{-1}p(t), \\ \dot{p}(t) = -\nabla E_{pot}(q(t)) - \gamma M^{-1}\tilde{p}(t) - \sqrt{2\beta^{-1}\gamma}\dot{w}(t), \end{cases}$$

where $(w(t))_{t \geq 0}$ is a Brownian motion on \mathbb{R}^{6N} , so that $\dot{w}(t)$ is a white noise force, and $\gamma > 0$. Rather than entering into technical details about diffusion processes, SDEs, and stochastic calculus in general, let us simply say that this process is the limit as δ vanishes of the following Euler scheme with time step δ :

$$\begin{cases} q_{k+1} - q_k = \delta M^{-1}p_k, \\ p_{k+1} - p_k = -\delta \nabla E_{pot}(q_k) - \delta \gamma M^{-1}p_k + \sqrt{2\beta^{-1}\gamma\delta}W_k, \end{cases}$$

where $(W_k)_{k \in \mathbb{N}}$ is a sequence of independent random variables of dimension $3N$ distributed according to the standard (i.e., mean zero, variance equal to the identity matrix) Gaussian distribution. In fact, rather than this naive Euler scheme, higher-order methods are used in practice, as we will see below. Similarly to the Euler scheme, these schemes require at each step the computation for a configuration q of the forces $-\nabla E_{pot}(q)$. The rest of this work is based on a decomposition of the forces of the form

$$\nabla_q E_{pot}(q) = \sum_{i=0}^K F_i(q),$$

for $K \geq 1$ and some vector fields F_i . Suppose that F_0 gathers short-range forces, typically cheap to compute (since each particle only interacts with a small number of neighbors through these forces) but fast-varying and high (in molecular dynamics, short-range forces are repulsive and singular at 0, for instance, Lennard–Jones forces scale as $1/r^{13}$, where r is the distance between the centers of two atoms), while the F_i 's for $i \geq 1$ are larger-range forces, numerically more intensive than F_0 (since an atom basically interacts with all the others through these forces), but also smaller and more regular. A natural idea is thus not to compute the long-range forces at each step or, in other words, to use different time steps for the different forces. Indeed, by classical considerations on the convergence of the Euler schemes, the time step should be related to the norm of the forces. This idea leads to the so-called multi-timestep methods, see Refs. 20 and 21 and references within.

B. Introduction to jump processes

The method developed in this work is different. We will design a continuous-time process based on the Langevin diffusion

$$\begin{cases} \dot{q}(t) = M^{-1}p(t), \\ \dot{p}(t) = -F_0(q(t)) - \gamma M^{-1}\tilde{p}(t) - \sqrt{2\beta^{-1}\gamma}\dot{w}(t), \end{cases} \quad (2)$$

to which random jumps (or collisions, or bounces) will be added that will take into account the forces F_i 's, $i \geq 1$, in such a way that ergodicity with respect to μ is enforced. Then, a unique time step will be used for the discretization. The resulting discrete-time chain will be such that the forces F_i 's are only computed at random steps.

Before giving a rigorous definition of the process, let us give a brief and informal definition of the jump mechanisms. Denote $v = M^{-1}p$ the velocities of the particles. The trajectory basically follows (2) except that, between times t and $t + \delta$ for δ small, the process has a probability $\lambda_i(q(t), v(t))\delta + o(\delta)$ to undergo a collision due to the i th force for all $i \in \llbracket 1, K \rrbracket$, where λ_i is called the i th jump rate and, here, is given by

$$\lambda_i(q, v) = \beta \max(0, v'F_i(q)). \quad (3)$$

At such a collision, the velocity jumps from v to

$$R_i(q, v) = v - 2 \frac{v'F_i(q)}{F_i(q)'M^{-1}F_i(q)} M^{-1}F_i(q). \quad (4)$$

Note that this is not defined if $F_i(q) = 0$, but in that case, the probability to see a collision is zero. When M is a scalar matrix, this means v is reflected orthogonally to $F_i(q)$. More generally, $v_* = R_i(q, v)$ is the only vector that conserves the kinetic energy (i.e., such that $v_*'Mv_* = v'Mv$) such that $v_*'F_i(q) = -v'F_i(q)$. In other words, $M^{1/2}v_*$ is the orthogonal reflection of $M^{1/2}v$ with respect to $M^{-1/2}F_i(q)$.

This kind of velocity jump mechanism has received interest over the past years in various communities concerned with sampling problems.^{22–24} The novelty in this work is that it is integrated within a classical Langevin diffusion through a force splitting and implemented in a practical MD code.

III. THEORETICAL SETTINGS

A. The theoretical continuous-time process

Let $T_0 = 0$ and suppose by induction that (q_t, p_t) has been defined for $t \in [0, T_n]$ for some $n \in \mathbb{N}$. We call T_n the n th jump (or collision) time. Let $(\tilde{p}, \tilde{q})_{t \geq T_n}$ be the solution of the SDE (2) with initial condition $(\tilde{p}(T_n), \tilde{q}(T_n)) = (p(T_n), q(T_n))$. Let $(\mathcal{E}_i)_{i \in \llbracket 1, K \rrbracket}$ be independent random variables with standard exponential distribution, i.e., such that $\mathbb{P}(\mathcal{E}_i > s) = \exp(-s)$ for all $s > 0$, independent from $(w(t))_{t \geq 0}$. For $i \in \llbracket 1, K \rrbracket$, let

$$S_i = \inf \left\{ t > T_n, \mathcal{E}_i < \int_{T_n}^t \lambda_i(\tilde{q}_s, M^{-1}\tilde{p}_s) ds \right\}, \quad (5)$$

where λ_i is given by (3). The next jump time is then defined as $T_{n+1} = \min\{S_i, i \in \llbracket 1, K \rrbracket\}$. Let I_{n+1} be the index such that $T_{n+1} = S_{I_{n+1}}$ (since almost surely $T_i \neq T_j$ for $i \neq j$, I_{n+1} is uniquely defined). For $t \in [T_n, T_{n+1})$, set $(p(t), q(t)) = (\tilde{q}(t), \tilde{p}(t))$. At time T_{n+1} , the position is continuous, i.e., $q(T_{n+1}) = \tilde{q}(T_{n+1})$, while the velocity $v = M^{-1}p$ undergoes a collision due to the I_{n+1}^{th} jump mechanism and jumps to the value

$$v(T_{n+1}) = R_{I_{n+1}}(\tilde{q}(T_{n+1}), \tilde{v}(T_{n+1})),$$

with R_i given by (4). The process is thus defined up to T_{n+1} and, by induction, up to T_k for all $k \in \mathbb{N}$. From now on, we assume that $\|F_i\|_\infty := \sup_{q \in \mathbb{R}^{3N}} \|F_i(q)\| < +\infty$ for all $i \in \llbracket 1, K \rrbracket$ (while F_0 may be unbounded and admit singularities). Under this assumption, using the bounds on the F_i 's and the fact the norm of the velocity is not modified at jump times, it can be proven that there is almost surely a finite number of jumps in a finite interval.²⁵ As a consequence, the trajectory is defined for all times.

It is readily checked²⁶ that $\int \mathcal{L}\phi d\mu = 0$ for all nice ϕ , where \mathcal{L} is the Markov generator associated with the process, which informally means that μ is an invariant distribution for the process. The rigorous proof that the process is ergodic with respect to μ in the typical settings of molecular dynamics (i.e., with singular forces) is beyond the scope of the present paper, but we give a sketch of the proof of the following:

Theorem 1. *In the case of periodic conditions, i.e., $q \in (\mathbb{R}/\mathbb{Z})^{3N}$, suppose that F_i is C^2 for all $i \in \llbracket 0, K \rrbracket$. Then, the process is ergodic with respect to μ , in the sense that (1) holds for all bounded measurable ϕ .*

Proof. It is well-known and easily checked (see Ref. 27 and references within) that the Langevin diffusion (2) admits E_{kin} as a Lyapunov function. Moreover, by hypoellipticity, its transition kernel admits for all positive time a smooth density with respect to the Lebesgue measure. Since the kinetic energy is unchanged at a jump time, E_{kin} is still a Lyapunov function for the jump process. Since the jump rates are bounded, there is for all positive time t a positive probability r (independent from the starting point of the process) that no jump occurs in the time interval $[0, t]$. Thus, the transition kernel of the jump process is bounded below by r times the transition kernel of the diffusion (2) and hence is bounded below (uniformly in the starting point in a given compact set) by constant times the Lebesgue measure on some compact set. This means a local Doeblin condition is satisfied, which, together with the drift condition enforced by the Lyapunov function, classically concludes (see Refs. 26–29 for details). \square

B. Interpretations

Let us give some intuitions on the meaning of the jump mechanisms. Suppose to fix ideas that $F_i = \nabla E_i$ for some potential E_i , $\beta = 1$, and M is the identity matrix. Then, from $\dot{q} = v$, we get $\partial_t E_i(q) = v'F_i(q)$; hence, $\lambda_i(q, v) = \max(0, \partial_t E_i(q))$. As a consequence,

$$\int_0^t \lambda_i(q(s), v(s)) ds = E_i(q(t)) - E_i(q(0)),$$

while E_i is increasing along the trajectory, and zero otherwise. The definition of the time S_i can thus be interpreted as follows: the exponential random variable \mathcal{E}_i drawn at the beginning of the trajectory is the total amount of energy barrier that the process is allowed to cross uphill. When all this budget is spent, the process jumps (see Fig. 1). Note that, due to the so-called lack of memory of the exponential law, i.e., the fact that, for all $s, t > 0$,

$$\mathbb{P}(\mathcal{E} > t + s \mid \mathcal{E} > t) = \mathbb{P}(\mathcal{E} > s),$$

it is in fact not necessary to keep in memory the accumulated jump rate involved in (5). Indeed, if we run the trajectory for some time

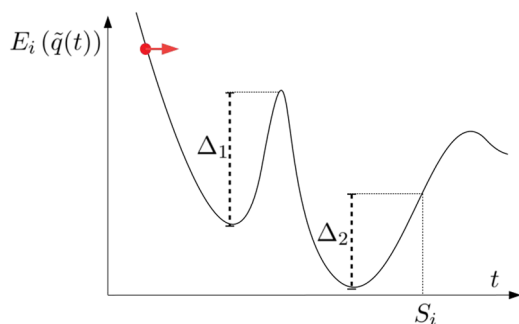


FIG. 1. As long as the process goes down the potential, there is no jump. When the total energy barrier crossed (here, $\Delta_1 + \Delta_2$) reaches the random value \mathcal{E}_i , the process jumps.

$t_0 > 0$ and no jump occurs, we can reset the accumulated jump rate to zero and draw new variables $\mathcal{E}_1, \dots, \mathcal{E}_k$, and this will not affect the statistics of the trajectory. In other words, the process is Markovian (at any time, the law of the future trajectory depends on the current position but not on the past trajectory).

If the process has reached (q, \tilde{v}) at time S_i and jumps, then $F_i(q) = \nabla E_i(q)$ is the normal vector at point q of the level set $\{E = E(q)\}$. Then, the jump from \tilde{v} to $R_i(q, \tilde{v})$ corresponds to a specular reflection on the level set (see Fig. 2).

Another interpretation is to see the process as the continuous limit of a lifted Metropolis–Hastings scheme, as in Refs. 22 and 26. For a given time step $\delta > 0$, consider the Markov transition from (q_n, v_n) to $(q_n + \delta v_n, v_n)$ with probability $\min(1, \exp(-E_i(q_n + \delta v_n) + E_i(q_n)))$, and to $(q_n + \delta(v_n + R_i(q_n, v_n)))/2, R_i(q_n, v_n)$ otherwise. Then, this step (approximately as δ vanishes) fixes the probability law μ , by design of the Metropolis–Hastings acceptance probability. Note that

$$\begin{aligned} \min\left(1, e^{-E_i(q+\delta v)+E_i(q)}\right) &= e^{-\max(0, E_i(q+\delta v)-E_i(q))} \\ &= e^{-\delta \max(0, v'F_i(q))} + o(\delta). \end{aligned}$$

Then, a bounce for the continuous process corresponds to a Metropolis rejection for the discrete chain. However, contrary to

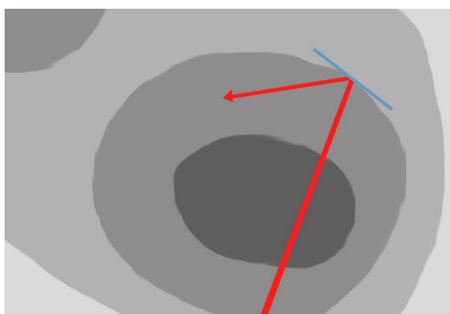


FIG. 2. At a collision time, the process bounces on the level set of the potential.

what happens for the classical reversible Metropolis–Hastings algorithm, here a rejection does not mean that the process is stopped (which would impair the exploration of the space and hence the variance of the ergodic estimation), only that it changes its direction. For more complete and rigorous considerations on this interpretation, we refer to Ref. 26 and references within.

C. Markov generator and splitting scheme

For a given observable φ , denote

$$P_t \varphi(q, p) = \mathbb{E}[\varphi(q_t, p_t) \mid (q_0, p_0) = (q, p)]$$

as the average of φ over the distribution of (q_t, p_t) . Then, this distribution is characterized by $\{P_t \varphi, \varphi \text{ continuous bounded}\}$. Similarly, for a Markov process, the motion of the process is characterized by the evolution of $P_t \varphi$ for a sufficiently large class of observables φ . This evolution is guided by the generator \mathcal{L} of the process, which is a linear operator defined by

$$\mathcal{L} \varphi = \lim_{t \rightarrow 0} \frac{1}{t} (P_t \varphi - \varphi),$$

for all suitable observable φ . In that case, from the Markovian property of the dynamics, we get that $\partial_t P_t \varphi = \mathcal{L} P_t \varphi$ for all $t \geq 0$ so that, informally, $P_t \varphi = e^{t\mathcal{L}} \varphi$. Here, for the process introduced in Sec. III A, following Refs. 25 and 26 (and the notations of Ref. 9), we get that $\mathcal{L} = \mathcal{L}_A + \mathcal{L}_B + \mathcal{L}_O + \mathcal{L}_J$, where

$$\begin{aligned} \mathcal{L}_A \varphi(q, p) &= (M^{-1} p) \nabla_q \varphi(q, p), \\ \mathcal{L}_B \varphi(q, p) &= -F_0(q) \nabla_p \varphi(q, p), \\ \mathcal{L}_O \varphi(q, p) &= -\gamma (M^{-1} p) \nabla_p \varphi(q, p) + \gamma \beta^{-1} \Delta_p \varphi(q, p), \\ \mathcal{L}_J \varphi(q, p) &= \sum_{i=1}^K \lambda_i(q, M^{-1} p) (\varphi(q, p_*) - \varphi(q, p)), \end{aligned}$$

with $p_* = MR_i(q, M^{-1} p)$, respectively, corresponds to the free transport (A), the forces (B), the friction/dissipation (O), and the jumps (J). We approximate the continuous-time dynamics above following the Trotter/Strang splitting scheme^{9,30}

$$e^{t\mathcal{L}} = e^{\frac{1}{2}t\mathcal{L}_B} e^{\frac{1}{2}t\mathcal{L}_J} e^{\frac{1}{2}t\mathcal{L}_A} e^{t\mathcal{L}_O} e^{\frac{1}{2}t\mathcal{L}_A} e^{\frac{1}{2}t\mathcal{L}_J} e^{\frac{1}{2}t\mathcal{L}_B} + o(t^2).$$

In other words, starting from the BAOAB scheme of Ref. 9, we add two half-time jump steps between the forces and the transport parts. The process corresponding to each (half) step can be exactly simulated. Let us focus in Sec. III D on the case of \mathcal{L}_J . The stochastic process (q_t, p_t) associated with \mathcal{L}_J is the following. The position $q_t = q$ is constant, and the velocity $v_t = M^{-1} p_t$ is a Markov chain that jumps from v to $R_i(q, v)$ at rate $\lambda_i(q, v)$ for all $i \in \llbracket 1, K \rrbracket$. Note that computing $\lambda_i(q, v)$ and $R_i(q, v)$ requires to compute $F_i(q)$.

D. Efficient sampling of a Markov chain

In a general abstract framework, let $\lambda_1, \dots, \lambda_K : \mathbb{R}^d \rightarrow \mathbb{R}_+$ be some jump rates and $R_1, \dots, R_K : \mathbb{R}^d \rightarrow \mathbb{R}^d$. Let us construct a Markov chain $X(t) \in \mathbb{R}^d$ that jumps from x to $R_j(x)$ at rate $\lambda_j(x)$ for $j \in \llbracket 1, K \rrbracket$. The basic definition of such a chain is the following: starting at $X(0) = x$, let $\mathcal{E}_1, \dots, \mathcal{E}_K$ be independent standard exponential random variables and $S_j = \mathcal{E}_j / \lambda_j(x)$. Then, $T = \min_{j \in \llbracket 1, K \rrbracket} S_j$ is

the next jump time, and the Markov chain jumps at time T from x to $R_J(x)$, where J is the index such that $T = S_J$ (almost surely uniquely defined). Then, the process starts again from this new position.

Suppose that there exist constants $\lambda_1^*, \dots, \lambda_K^* > 0$ such that $\lambda_j(x) \leq \lambda_j^*$ for all $x \in \mathbb{R}^d$, $j \in \llbracket 1, K \rrbracket$. Then, the construction above is equivalent (in the sense that it defines the same Markov chain) to the following. Starting at $X(0) = x$, let $\mathcal{E}_1, \dots, \mathcal{E}_K$ be independent standard exponential random variables and $S_j = \mathcal{E}_j/\lambda_j^*$. Let $T = \min_{j \in \llbracket 1, K \rrbracket} S_j$ and J be the index such that $T = S_J$. We say that at time T , a jump of type J is proposed. Let U be a random variable uniformly distributed over $[0, 1]$. If $U \leq \lambda_j(x)/\lambda_j^*$ [which happens with probability $\lambda_j(x)/\lambda_j^*$], then the chain jumps from x to $R_J(x)$ at time T (we say that the jump is accepted). Otherwise, the chain does not jump, i.e., $X(T) = x$ (we say that the jump is rejected). In both cases, the process starts again from its current position.

From a computational point of view, the main advantage of the second method is that the rate $\lambda_j(x)$ only has to be computed when a jump of type j is proposed.

For a given fixed $t > 0$, let M_j be the number of proposed jumps of type j in the time interval $[0, t]$ and $\{T_{k,j}\}_{k \in \llbracket 1, M_j \rrbracket}$ be the times of these proposals. Classical properties of the exponential distribution ensure the following facts: M_j follows a Poisson distribution with parameter $\lambda_j^* t$ and, conditionally to M_j , the proposal times $\{T_{k,j}\}_{k \in \llbracket 1, M_j \rrbracket}$ are independent and uniformly distributed over $[0, t]$. Moreover, if $i \neq j$, then M_j and M_i are independent and so are $\{T_{k,i}\}_{k \in \llbracket 1, M_i \rrbracket}$ and $\{T_{k,j}\}_{k \in \llbracket 1, M_j \rrbracket}$. In fact, we do not need to know the values of the proposal times $T_{k,j}$ since the process is constant between two proposals, but we only need to know the order in which the different jump types are proposed.

As a consequence, we can sample $X(t)$ as follows: First, draw M_1, \dots, M_K as independent Poisson variables with respective parameters $\lambda_j^* t$ (that are, for each jump type, the total number of jumps proposed during the time interval $[0, t]$). Then, draw a jump type $J \in \llbracket 1, K \rrbracket$ in such a way that $\mathbb{P}(J = j) = M_j/(M_1 + \dots + M_K)$ for all $j \in \llbracket 1, K \rrbracket$. Propose a jump of type J , namely: with probability $\lambda_J(x)/\lambda_J^*$, x jumps to $R_J(x)$; otherwise, x stays at x . Then, update the remaining number of type J jump proposals; i.e., set $M_J \leftarrow M_J - 1$. Repeat until all the jump proposals have been considered, i.e., $M_1 = \dots = M_K = 0$. Then, the current position is $X(t)$.

The method is efficient if the upper bounds λ_j^* are small, so that the number of jump proposals [and thus the number of computations of $\lambda_j(x)$] is small. Of course, for λ_j^* to be small, we need at least the rates to be small and then the upper bound to be close to these rates.

IV. A FIRST EXAMPLE

A. Decomposition of the forces

Here, we consider a total potential energy

$$E_{pot}(q) = E_{elec}(q) + E_{vdW}(q) + E_{bond}(q),$$

with, respectively, the Coulomb electric, van der Waals, and bond potential. For the sake of simplicity, in this first example, the jump mechanisms are only used for the long-range van der Waals forces.

The van der Waals potential is

$$E_{vdW}(q) = \sum_{i \neq j} W(|q_i - q_j|),$$

where $q_i \in \mathbb{R}^3$ are the coordinates of the center of the i th atom, $i \in \llbracket 1, N \rrbracket$, and

$$W(r) = E_0 \left(\left(\frac{r_0}{r} \right)^{12} - \left(\frac{r_0}{r} \right)^6 \right) \quad (6)$$

for some parameters $E_0, r_0 > 0$. Consider some smooth switching function χ from \mathbb{R}_+ to $[0, 1]$ with $\chi(r) = 0$ if $r < r'$ and $\chi(r) = 1$ if $r > r''$ for some thresholds $0 < r' < r''$. For $i \in \llbracket 1, N \rrbracket$, let A_i be a $(3N) \times 3$ matrix with all coefficients equal to zero except $A(3(i-1) + 1, 1) = A(3(i-1) + 2, 2) = A(3(i-1) + 3, 3) = 1$. In other words, if $v = (v_1, \dots, v_N) \in \mathbb{R}^{3N}$ are the velocities of the N atoms, then $A_i'v = v_i \in \mathbb{R}^3$. Then, denoting $r_{ij} = |q_i - q_j|$, decompose

$$\begin{aligned} \nabla E_{vdW}(q) &= \sum_{i=1}^N A_i \nabla_{q_i} E_{vdW}(q) \\ &= \sum_{i=1}^N A_i \sum_{j \neq i} (1 - \chi_1(r_{ij}) + \chi_1(r'_{ij})) \nabla_{q_i} (W(r_{ij})) \\ &= F_{shortVdW}(q) + \sum_{i=1}^N A_i \sum_{j=1}^N F_{ij}(q), \end{aligned}$$

with

$$\begin{aligned} F_{shortVdW}(q) &= \sum_{i=1}^N A_i \sum_{j \neq i} (1 - \chi(r_{ij})) \nabla_{q_i} (W(r_{ij})), \\ F_{ij}(q) &= \chi(r_{ij}) \nabla_{q_i} (W(r_{ij})) \end{aligned}$$

for $j \neq i$, and zero otherwise. Note that we applied the switching function directly at the gradient level and that the short-range van der Waals forces $F_{shortVdW}$ are numerically less intensive than the long-range ones.

As a conclusion, decompose the forces as

$$\nabla E_{pot}(q) = F_0 + \sum_{i=1}^N A_i \sum_{j=1}^N F_{ij}(q),$$

with

$$F_0(q) = \nabla E_{elec}(q) + F_{shortVdW}(q) + \nabla E_{bond}(q).$$

Note that $v' A_i F_{i,j} = v_i' F_{i,j}$. It means that the jump mechanisms associated with the vector field $A_i F_{i,j}$ only involve v_i . As a consequence, with such a decomposition, a Markov chain $V = (V_1, \dots, V_N) \in \mathbb{R}^{3N}$ with generator given by \mathcal{L}_J (for fixed positions q) is such that the V_i 's are independent Markov chains that can be simulated in parallel. For all $i \in \llbracket 1, N \rrbracket$, V_i is a Markov chain that jumps from v_i to $R_{i,j}(v_i)$ at rate $\lambda_{i,j}(v_i)$ for $j \in \llbracket 1, N \rrbracket$, where

$$\begin{aligned} \lambda_{i,j}(v_i) &= \beta \max(0, v_i' F_{i,j}(q)), \\ R_{i,j}(v_i) &= v_i - 2 \frac{v_i' F_{i,j}(q)}{\|F_{i,j}(q)\|^2} F_{i,j}(q), \end{aligned}$$

(the mass of an atom is a scalar matrix, and thus, it disappears in the expression of R). According to Sec. III D, we need to find an upper

bound of the jump rates. Here,

$$\lambda_{ij}(v_i) \leq \beta \|v_i\| \|F_{ij}(q)\|.$$

Note that, at a jump time, the velocity v_i is transformed by an orthogonal reflection, so that its norm is conserved and thus $\|V_i(t)\|$ is constant along the trajectory of the Markov chain. On the other hand, using that $\|\nabla_{q_i}(r_{ij})\| = 1$ and $\|\chi\|_\infty = 1$, we get that

$$\|F_{ij}(q)\| \leq \|\chi(r_{ij})W'(r_{ij})\| \leq \|\chi W'\|_\infty.$$

As a consequence, given any bound L of $\|\chi W'\|_\infty$, we bound the jump rate by

$$\lambda_{ij}(q, v) \leq \lambda_{ij}^* := \beta \|v_i\| L.$$

We can bound χ by the indicator function of $[r'', +\infty[$ to analytically derive an explicit bound of $\|\chi W'\|_\infty$. Nevertheless, as we saw in Sec. III D, the lower is our bound, the more efficient is the algorithm, so instead of using an explicit non-optimal bound, it is better to numerically compute $\|\chi W'\|_\infty$ once and for all at the beginning of the simulation. This is easily done since $\chi W'$ is a one-dimensional function.

B. The algorithm

Keep the notations of Sec. IV A, and denote δ the time step. The symmetric splitting scheme presented in Sec. III C reads

- (B) Set $p \leftarrow p - \frac{1}{2}\delta F_0(q)$.
- (J) Set $p \leftarrow MV(\delta/2)$, where $V(0) = M^{-1}p$, and $V = (V_1, \dots, V_N)$, where for all $i \in \llbracket 1, N \rrbracket$, $(V_i(t))_{t \geq 0}$ is a Markov chain that jumps from v_i to $R_{i,j}(v_i)$ at rate $\lambda_{i,j}(v_i)$ for all $j \in \llbracket 1, N \rrbracket$ (see below).
- (A) Set $q \leftarrow q + \frac{1}{2}\delta M^{-1}p$.
- (O) Set $p \leftarrow e^{-\gamma\delta M^{-1}} p + \sqrt{\beta^{-1}(1 - e^{-2\gamma\delta M^{-1}})} MG$ with G a standard Gaussian random variable.

Then, repeat steps A, J, and B in this order. It remains to describe in detail the step J, which is the following:

- For $i \in \llbracket 1, N \rrbracket$, do
- Initialize $v_i = \frac{1}{m_i} p_i$.
- Draw M_i a Poisson random variable with parameter $N\beta \|v_i\| L\delta/2$.
- For $k = \llbracket 1, M_i \rrbracket$, do
- Draw J and U uniformly distributed, respectively, over $\llbracket 1, N \rrbracket$ and $[0, 1]$
- If $U < \lambda_{i,J}(v_i)/\lambda_{i,J}^*$, do
- $v_i \leftarrow R_{i,J}(v_i)$
- end if
- end do
- end do.

Note the following slight modification with respect to the general settings of Sec. III D: we do not generate different Poisson variables $M_{i,j}$ for each $j \in \llbracket 1, N \rrbracket$ corresponding to the N jump mechanisms involving v_i . The reason is that the bounds $\lambda_{i,j}^*$ are the same for all j , and thus, we directly sample the total number of jumps proposed for the i th velocity (hence the N factor in the parameter of the Poisson variable M_i) and, for each of these jump proposals,

we chose at random its type J (again, since the bounds are the same, the index J such that $S_J = \min_{j \in \llbracket 1, N \rrbracket} S_j$ is uniformly distributed over $\llbracket 1, N \rrbracket$).

C. Computational gain

In a usual scheme (for instance, the BAOAB one⁹), at each time step, $\nabla_q E_{VdW}$ has to be computed, which demands $N(N-1)$ computations of quantities of the form $\nabla_{q_i}(W(r_{ij}))$. Let us compare this with our algorithm.

At each step, $F_{shortVdW}$ has to be computed once, which demands $\sum_{i=1}^N N_{n(i)}$ similar computations, where $N_{n(i)}$ is the number of atoms in a neighbor list that contains at least all atoms at distance less than r'' of the i th one. Moreover, at each half time step and for each atom $i \in \llbracket 1, N \rrbracket$, M_i computations are required, whose expected value is $\beta N \|v_i\| L\delta/2$. At equilibrium, v_i is distributed according to an isotropic Gaussian law with variance $1/(m_i\beta)$, where m_i is the mass of the i th atom. By the Jensen inequality, the expected value of $\|v_i\|$ is less than the standard deviation of this distribution, $\sqrt{3/(m_i\beta)}$. Thus, the average total number of computations of quantities of the form $\nabla_{q_i}(W(r_{ij}))$ at each time step is of order

$$\sum_{i=1}^N N_{n(i)} + \sqrt{3\beta N L \delta} \sum_{i=1}^N m_i^{-1/2}.$$

Note that the two parts of this quantity depend (in an opposite way) on the choice of the radius r'' that distinguishes short- and long-range interaction. Indeed, as r'' increases, $N_{n(i)}$ increases but the Lipschitz constant L decreases. The optimal choice of r'' should minimize the sum.

If we suppose that $N_{n(i)} \ll N$ when N becomes large (the other parameters being fixed), which is the case if the volume of the system increases with a constant density, then the cost of computing the short-range forces is negligible with respect to N^2 . As a consequence, for large N , the expected computation gain, in terms of number of computations of quantities of the form $\nabla_{q_i}(W(r_{ij}))$, from using bounces instead of computing the full gradient at each step is a factor

$$\alpha := \sqrt{3\beta L \delta} \frac{1}{N} \sum_{i=1}^N m_i^{-1/2}.$$

The parameters of the droplet numerical experiments of Sec. VI yield $\alpha = 5.1 \times 10^{-7}$ for a box of 1500 atoms water box using TIP3P model. In other words, concerning the computations of the long-range van der Waals forces, the cost is improved by five orders of magnitude.

Note that αN^2 , which is the numerical cost of the jumps at each step, is proportional to the time step δ . This means that this cost by unit of (simulation) time is in fact independent from the time step. In other words, the number of jumps proposed in a given time interval is independent from the time step, which is consistent with the extensivity property of the Poisson processes. The time between two jump proposals (and thus two computations of the force) can be seen as a random time step. This inter-jump time is, by design, automatically adapted to the Lipschitz norm of the corresponding force (each force being treated independently from the others). However, note that, even though the numerical cost of a given simulation is independent from the time step (as far as the jump mechanism is

concerned), the quality of the result is still impacted by this time step, since the Trotter splitting induces some error with respect to the continuous-time theoretical process.

V. THE FINAL “BOUNCE” ALGORITHM

In Sec. IV, the algorithm was kept simple for the sake of clarity. We now present the full algorithm that we will hereafter denote as BOUNCE. We will apply the approach to standard biomolecular force fields: AMBER³¹ and Chemistry at Harvard Molecular Mechanics (CHARMM)³² in the context of two types of boundary conditions: droplets where all the interactions are computed in real space and periodic boundary conditions (PBC) using standard smooth particle mesh Ewald (SPME) method.³³

First, two details regarding the van der Waals forces have been eluded in Sec. IV C: in practice, (6) is already multiplied by a switching function (there is no van der Waals interaction between two particles that are too far). In particular, when periodic boundary conditions are applied, the cutoff is such that an atom is never interacting with multiple periodic replicas since the cutoff is smaller than half the size of the box (minimum image convention). For a given particle i , the number N in the parameter of the Poisson law that gives the total number M_i of proposed jumps is not the total number of particles but the total number of particles that interact with i through the van der Waals force (and, when a particle J is drawn to determine the type of the jump, J is uniformly drawn between these particles, not all $[[1, N]]$). Moreover, the parameters E_0 and r_0 of the interaction depend on the nature of the two particles involved, and thus, the cutoff parameter of χ and the Lipschitz bound L may depend on the different types of particles.

In the droplet simulations, the Coulomb interaction is treated as the van der Waals one: a cutoff of 5 Å determines which interaction falls within short range and long range, the short range being evaluated in a standard way at each time step and the long range being evaluated with the same jump mechanism as described above, after having determined the proper bounds. A switching function whose characteristics are detailed in [supplementary material](#) is applied at the gradient level so that the short-range Coulomb force goes smoothly to zero near the cutoff. Note that due to the slow decay of Coulomb interactions with respect to the interatomic distance, no cutoff is applied to the full Coulomb interaction.

For periodic boundary conditions with the SPME method, Coulomb interactions are classically divided into a direct part and a reciprocal part,

$$E_{elec}(q) = \sum_{i \neq j} \frac{\epsilon_{ij}}{r_{ij}} [\text{erf}(r_{ij}) + (1 - \text{erf}(r_{ij}))] \\ := E_{recip}(q) + E_{direct}(q)$$

with erf being the error function. Following the SPME algorithm, the reciprocal part is computed in the Fourier space with fast Fourier transform. We decompose the direct part as

$$\nabla E_{direct}(q) = \sum_{i \neq j} [\chi(r_{ij}) + (1 - \chi(r_{ij}))] \nabla \left(\frac{\epsilon_{ij}}{r_{ij}} (1 - \text{erf}(r_{ij})) \right) \\ = F_{shortdir}(q) + \sum_{i \in N} A_i \sum_{j \neq i} \tilde{F}_{ij}(q),$$

for some cutoff function χ , like for the van der Waals forces. The long-range parts \tilde{F}_{ij} are Lipschitz vector fields that are treated through a jump mechanism.

The short-range van der Waals and Coulomb forces, together with the bond forces ∇E_{bond} , are all gathered in F_0 . The last forces that have to be taken into account are the reciprocal forces ∇E_{recip} , which are many-body forces. In the basic BOUNCE algorithm, they are also integrated in F_0 and hence evaluated at each time step. In fact, there would be no theoretical objection to the use of jump mechanisms to treat many-body forces but, as we saw, the practical efficiency of the method is related to the capacity to get good (i.e., small) bounds on the forces (in particular, for each type of jump, the average number of proposals per time step should be less than 1 so that the forces are often not computed at all). This is a nontrivial question in the case of the reciprocal forces. Nevertheless, we can also use for them a classical multi-timestep method, the reference system propagator algorithm (RESPA).²⁰ More precisely, denote by Q the succession of steps of BJAQJB as defined in Sec. IV B, with some time step δ and where the jump step J involves both the van der Waals and Coulomb long-range forces. Let m be an even integer.

Then, the evolution of (q_t, p_t) over a time step $\Delta = m\delta$ is approximated as follows:

- Set $p \leftarrow -\frac{\Delta}{2} \nabla E_{recip}(q)$.
- Perform m times the step Q with time step δ .
- Set $p \leftarrow -\frac{\Delta}{2} \nabla E_{recip}(q)$.

In other words, a Trotter/Strang splitting scheme is performed with the decomposition

$$e^{\Delta \mathcal{L}} \simeq e^{\frac{\Delta}{2} \mathcal{L}_{recip}} \left(e^{\delta \mathcal{L}_Q} \right)^m e^{\frac{\Delta}{2} \mathcal{L}_{recip}},$$

where

$$\mathcal{L}_{recip} \varphi(q, p) = -\nabla E_{recip}(q) \nabla_p \varphi(q, p).$$

In practice, what we will call BOUNCE-RESPA in the rest of the text is a further standard bonded/non-bonded decomposition, where not only is the reciprocal-space part of the Coulomb interaction evaluated at an outer bigger time step, but also the rest of the non-bonded part of the potential is not evaluated through the jump mechanism,

$$F_{out} = F_{shortdir} + F_{shortVdW} - \nabla E_{recip}$$

such that the algorithm reads

$$e^{\Delta \mathcal{L}} \simeq e^{\frac{\Delta}{2} \mathcal{L}_{out}} \left(e^{\delta \mathcal{L}_Q} \right)^m e^{\frac{\Delta}{2} \mathcal{L}_{out}}.$$

The same reasoning can be made for the droplet simulations without the reciprocal-space part of the Coulomb energy and by replacing the short-range direct-space part of the Ewald sum by the actual short-range part of the total Coulomb energy. Note that in principle, the forces that are evaluated through jumps could be placed at the outer level of the splitting scheme but, due to the extensivity of the jumps, the computational efficiency would not improve, and we would expect the dynamics to be more perturbed by the splitting. As a consequence, the jump step remains a part of \mathcal{L}_Q .

As we see, the RESPA and BOUNCE methods are straightforwardly compatible. Note that while the basic BOUNCE algorithm avoids resonance problems since it is a simple discretization (with a unique time step δ) of an ergodic theoretical process, the BOUNCE-RESPA may, like the original RESPA, suffer from this limitation. Furthermore, it is straightforward to use an additional splitting of the potential like RESPA1 (with PME)³⁴ with the BOUNCE procedure: in this case, F_{out} is further split into a part corresponding to the short-range portion of the non-bonded term that is evaluated at an intermediate time step and a part consisting only of the reciprocal-space interactions that are evaluated at a larger outer time step.

VI. ASSESSMENTS OF BOUNCE CAPABILITIES: NUMERICAL EXPERIMENTS

To assess the computational capabilities and accuracy of the BOUNCE integrator, we have performed the first implementation in the Tinker-HP massively parallel software for molecular dynamics.¹ All tests have been performed using the classical, non-polarizable TIP3P³⁵ and Simple Point Charge (SPC)³⁶ water models and with the CHARMM³² or AMBER³¹ force fields. All bonds are flexible. Computations have been performed on the Occigen supercomputer (CINES, France) using Intel Broadwell processors. The software will be made freely available to academic users within the next release of the Tinker-HP code.^{1,37,38}

A. Evaluation of direct-space accelerations using BOUNCE: Proof-of-principle droplet simulations

Because the splitting of the potential between short and long range is straightforward in this context, pure direct-space simulations are a good proof of concept to assess the capabilities of this new integrator. Tests have been performed on large droplets of TIP3P water: first a droplet of 500 TIP3P water molecules and 9737 atoms solvated ubiquitin protein described with the AMBER (f99) force field at 300 K. In both cases, a repulsive van der Waals-like wall was imposed to block the atoms to drift away during the simulation. Table I shows the relative acceleration of BOUNCE compared to reference BAOAB (1 fs) simulations and to the popular RESPA with a bonded/non-bonded split (0.5 fs/2 fs) approach. As expected since BOUNCE evaluates much less often the long-range part of

electrostatic and van der Waals interactions than standard integrators, it is nearly four times faster than the BAOAB and twice as fast as BAOAB-RESPA approaches. In fact, when using BOUNCE, we observe an average number of jump proposals per time step per atom for the van der Waals interaction of 4.7×10^{-4} , in accordance with the *a priori* estimate of $\alpha \times N = 7.65 \times 10^{-4}$ given in Sec. IV C (corresponding to $\alpha = 5.1 \times 10^{-7}$), which explains that BOUNCE always provides a net computational advantage. In that connection, Table I also shows that the BOUNCE algorithm does adapt to each situation as different force field models give rise to different accelerations; i.e., TIP3P is slightly faster than SPC as is the AMBER/TIP3P combination compared to the CHARMM/TIP3P one. A straightforward analysis of the accuracy of the method can be performed by computing the average potential energies of the systems. For the pure TIP3P water droplet after 100 ps simulations, the average BAOAB total potential energy is of -4371.22 kcal/mol (-4390.31 kcal/mol with BAOAB-RESPA 0.5/2 fs), whereas BOUNCE obtains a close value of -4387.18 kcal/mol, with a reference of -4411.80 kcal/mol for Verlet 0.5 fs. In practice, the BOUNCE energy is closer to the Verlet reference than BAOAB-RESPA, which amounts for -4475.90 kcal/mol. Similar results are obtained for a non-homogeneous system such as the solvated ubiquitin (9737 atoms) (see Table I and supplementary material), demonstrating the robustness of the BOUNCE integrator.

B. Accelerating condensed-phase simulations: With BOUNCE-RESPA while keeping accuracy

At this stage, the application of BOUNCE to direct-space computations is promising and can already be useful in a variety of cases such as Quantum Mechanics/Molecular Mechanics (QM/MM) computations or continuum solvation models but we want to see how the method performs in the widely used context of periodic boundary conditions using the Particle Mesh Ewald method. This section proposes an in-depth study of the capabilities of the BOUNCE algorithm to reproduce condensed-phase properties of bulk water such as the average potential energies and radial distribution functions. Although the BOUNCE process is not designed to reproduce dynamical properties of the Langevin dynamics, we also include the diffusion coefficient since it is an indicator of the sampling rate.¹⁹ The rest of the section shows results on three test systems: first a cubic water box of edge 24.662 \AA , and then ubiquitin and DHFR proteins solvated in a $72 \times 54 \times 54 \text{ \AA}^3$ water and

TABLE I. Relative speedups for 1500 molecules TIP3P water box and solvated ubiquitin (AMBER FF99/CHARMM protein + TIP3P water) direct-space droplet systems simulation for the BAOAB, BOUNCE, and BAOAB-RESPA integrators, with BAOAB as a reference.

Water box	Water box	Water box
BAOAB	BAOAB-R(0.5 fs/2 fs)	BAOAB
1 (TIP3P)	1.98 (TIP3P)	3.96 (TIP3P)
1 (SPC)	1.95 (SPC)	3.82 (SPC)
Ubiquitin	Ubiquitin	Ubiquitin
BAOAB (1 fs)	BAOAB-R(0.5 fs/2 fs)	BAOAB
1 (AMBER)	1.97 (AMBER)	3.81 (AMBER)
1 (CHARMM)	1.95 (CHARMM)	3.61 (CHARMM)

TABLE II. Speedup comparison on 1500 atoms periodic boundary conditions water-box system (SPC and TIP3P water models) of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, BAOAB-RESPA, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

1 fs	0.5/2 fs	0.5/2/6 fs
BAOAB	BAOAB-R	BAOAB-R1
1.58 (SPC)	2.575 (SPC)	3.25 (SPC)
1.58 (TIP3P)	2.51 (TIP3P)	3.18 (TIP3P)
1 fs	0.5/2 fs	0.5/2/6 fs
BAOAB	BAOAB-R	BAOAB-R1
1.78 (SPC)	1.78 (SPC)	2.5 (SPC)
1.78 (TIP3P)	1.78 (TIP3P)	2.49 (TIP3P)

TABLE III. Speedup comparison on 9737 atoms periodic boundary conditions ubiquitin protein system (AMBER FF99/CHARMM model protein and TIP3P model water) of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, BAOAB-RESPA, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

1 fs	0.5/2 fs	0.5/2/6 fs
BOUNCE	BOUNCE-R	BOUNCE- R1
1.40 (AMBER)	1.85 (AMBER)	2.7 (AMBER)
1.67 (CHARMM)	2.2 (CHARMM)	3.2 (CHARMM)
	0.5/2 fs	0.5/2/6 fs
	BAOAB-R	BAOAB-R1
	1.64 (AMBER)	2.55 (AMBER)
	1.65 (CHARMM)	2.67 (CHARMM)

TABLE IV. Speedup comparison on 23 558 atoms DHFR periodic boundary conditions protein system (CHARMM force field and TIP3P water) of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, BAOAB-RESPA, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

1 fs	0.5/2 fs	0.5/2/6 fs
BOUNCE	BOUNCE-R	BOUNCE- R1
1.36	1.71	2.43
	0.5/2 fs	0.5/2/6 fs
	BAOAB-R	BAOAB-R1
	1.57	2.21

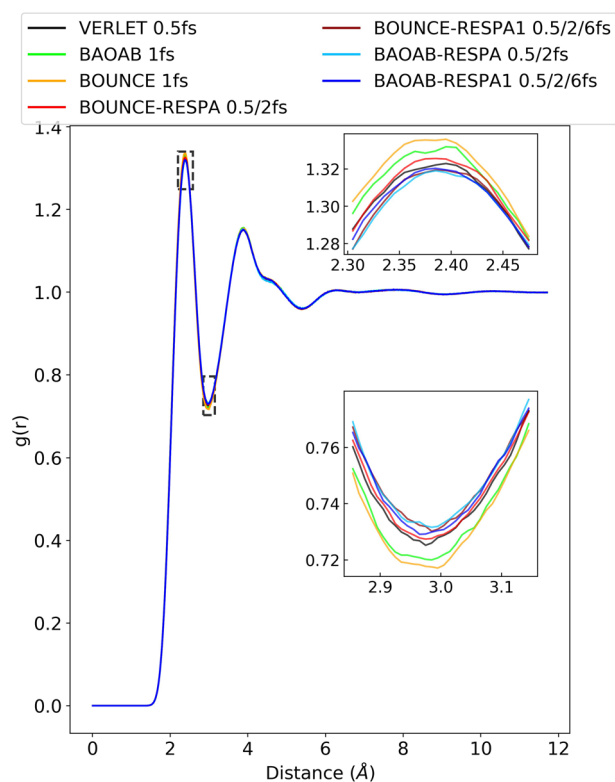
$64 \times 64 \times 64 \text{ \AA}^3$ boxes, respectively (see simulation details in [supplementary material](#)).

1. BOUNCE computational performance in periodic boundary conditions: Coupling with RESPA/RESPA1

We chose BAOAB (1 fs) with a friction of 1 ps^{-1} and Verlet (0.5) as a reference and compared BOUNCE to them along with BOUNCE-RESPA, BOUNCE-RESPA1, and BAOAB-RESPA1

TABLE V. Speedup comparison on 9737 atoms periodic boundary conditions ubiquitin protein system (CHARMM force field and TIP3P water) of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, BAOAB, BAOAB-RESPA, and BAOAB-RESPA1 integrators using a 10 \AA real-space cutoff and a 1.2 \AA grid spacing for PME, with BAOAB and a 7 \AA real-space cutoff and a 0.8 \AA grid spacing for PME (see [supplementary material](#)) as a reference.

1 fs	0.5/2 fs	0.5/2/6 fs
BOUNCE	BOUNCE-R	BOUNCE- R1
1.88	2.33	3.29
	0.5/2 fs	0.5/2/6 fs
	BAOAB-R	BAOAB-R1
	1.33	2.35

**FIG. 3.** Hydrogen-hydrogen radial distribution obtained with 1500 atoms periodic boundary conditions water box using the SPC model after a 2 ns simulation. Comparison of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

integrators. The BOUNCE-RESPA and BAOAB-RESPA integrators used a bonded/non-bonded split with a 0.5 fs time step for the bonded forces and a 2 fs for the non-bonded ones. The BOUNCE-RESPA1 and BAOAB-RESPA1 use the splitting described above and a 0.5 fs time step for the bonded forces, a 2 fs for the intermediate ones, and a 6 fs time step for the long-range ones. First, as expected, replacing BAOAB by BOUNCE always results in a net acceleration as the discussed average number of jump proposals per particle per time step of 5.8×10^{-3} is observed in practice for the direct-space interactions. BOUNCE alone without any reciprocal-space-specific treatment appears competitive with the popular RESPA with a bonded/non-bonded split (0.5/2 fs) approach, whereas all BOUNCE-RESPA or BOUNCE-RESPA1 outperforms their BAOAB counterparts (see [Tables II–IV](#)). It is important to note that by definition, different systems with different model parameters (TIP3P vs SPC, CHARMM vs AMBER) will result in different bounds in the algorithm and therefore BOUNCE will perform differently among them. It is worth mentioning here too that all proposed results depend on the splitting of the potential used to define the various blocks of the schemes, so that one could choose other cutoffs to define the portion of the potential treated by jumps and the range of the window switch between short and long range that could lead to

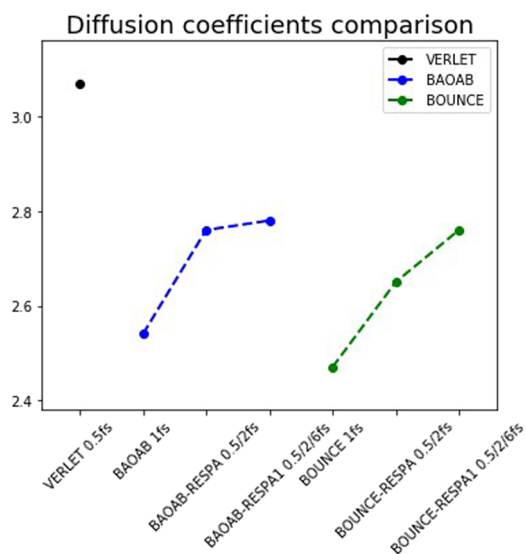


FIG. 4. Diffusion coefficient in 1500 atoms periodic boundary conditions water box using the SPC model after a 2 ns simulation. Comparison of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

improved performances. In order to illustrate that, we also compared the performance of simulations using the ubiquitin system and the CHARMM force field with a larger real-space cutoff of 10 Å (instead of 7 Å) and a correspondingly larger grid spacing of 1.2 Å (instead of 0.8 Å) as done in other codes.³ The results (see Table V) show an increase of the computational speedup from 1.67 to 1.88 for the pure BOUNCE integrator, 2.22 to 2.33 for the BOUNCE-RESPA, and 3.2 to 3.29 for the BOUNCE-RESPA1 integrator compared to the original BAOAB setup. Finally, another splitting between short range

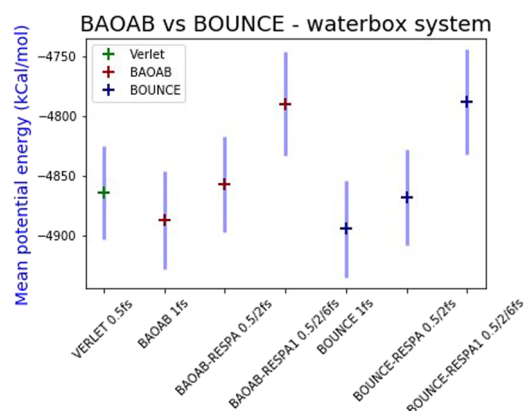


FIG. 5. Average potential energy comparison in 1500 atoms periodic boundary conditions water box using the SPC model after a 2 ns simulation. Comparison of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

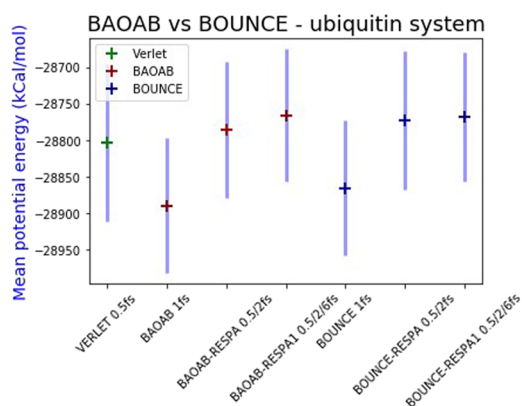


FIG. 6. Average potential energy comparison in 9737 atoms periodic boundary conditions solvated ubiquitin protein system (AMBER FF99 protein force field and TIP3P water) after a 1 ns simulation. Comparison of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

and long range for the Coulomb interaction could also be used and could lead to enhanced performances. Since testing all possibilities is beyond the scope of this initial paper, we chose the RESP A1 way of splitting the Coulomb interaction with the SPME method since

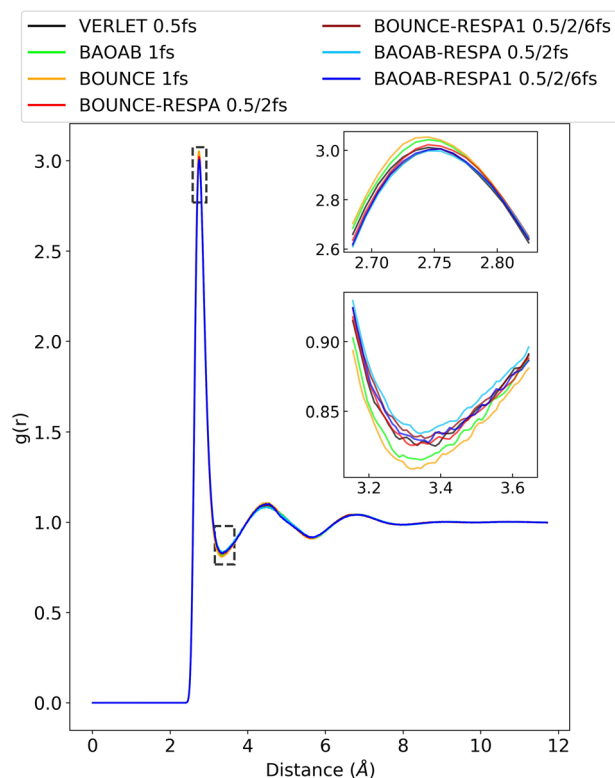


FIG. 7. Oxygen-oxygen radial distribution function obtained with 1500 atoms periodic boundary conditions water box using the SPC model after a 2 ns simulation. Comparison of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

its accuracy is less sensitive to the switching function between short and long range.³⁹

2. BOUNCE accuracy in periodic boundary conditions: Computation of condensed-phase properties

We first assessed the capability of the BOUNCE integrator to reproduce the radial distribution functions of water. Figure 7 displays the oxygen–oxygen radial distribution in 1500 atoms water box with the SPC model, Fig. 8 the oxygen–hydrogen radial distribution with the same system and model, and Fig. 3 the hydrogen–hydrogen one, after a 2 ns simulation. As a reference, we choose a velocity Verlet integrator with a small time step (0.5 fs) and the Bussi thermostat.⁴⁰ We compared it to the results obtained with BOUNCE along with BOUNCE-RESPA, BOUNCE-RESPA1, and the various BAOAB-based integrators. Figures 4–8 show that the native BOUNCE results are extremely close to the reference BAOAB and Verlet values. When coupled with RESPA or RESPA1, the BOUNCE approaches give a comparable agreement with the reference compared to their BAOAB-RESPA(1) counterparts. Similar trends are observed for average potential energies (see Figs. 5 and 6). Knowing that the continuous-time dynamics approached by BOUNCE

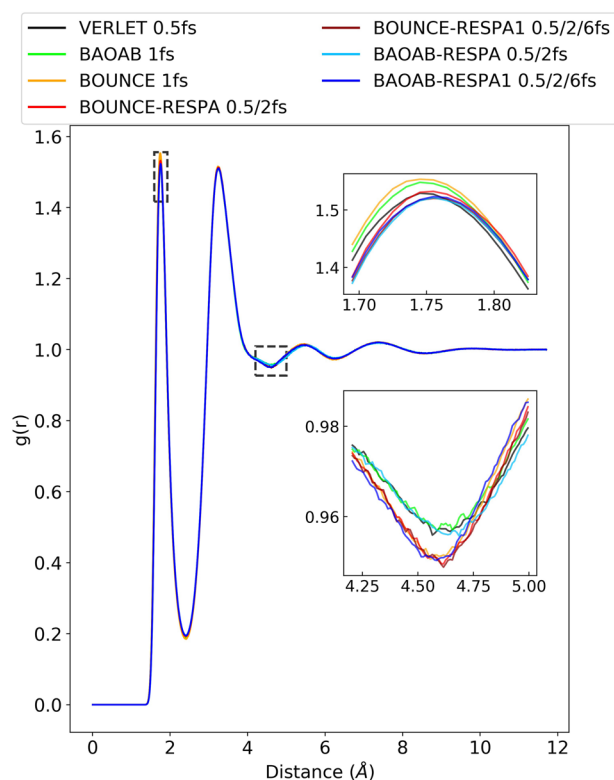


FIG. 8. Oxygen–hydrogen radial distribution obtained with 1500 atoms periodic boundary conditions water box using the SPC model after a 2 ns simulation. Comparison of the BOUNCE, BOUNCE-RESPA, BOUNCE-RESPA1, and BAOAB-RESPA1 integrators, with BAOAB as a reference.

is different than the Langevin one (given a fixed friction coefficient), which is in turn different than the microcanonical one, we expect dynamical properties computed with these approaches to differ. But the diffusion coefficient of a dynamic is also an indicator of its sampling rate,¹⁸ which makes it interesting to compute: a low self-diffusion coefficient would indicate a lower sampling rate and would then negate the observed computational speedups. Although BOUNCE does indeed affect the dynamics (see discussion in Sec. III A), the jumps only concern a small fraction of the forces and therefore the self-diffusion coefficient (that we computed using the Einstein relation) turns out to be very close to those of the initial Langevin dynamics. In practice, Fig. 4 shows that the self-diffusion constant of BOUNCE and its RESPA-based variations remains close to the reference BAOAB, meaning that using BOUNCE does not degrade the sampling rate of a dynamic compared to a pure Langevin dynamic computed with BAOAB and with same friction.

VII. CONCLUSION AND PERSPECTIVES

We introduced a new MD integrator, denoted as BOUNCE, in which the slowly varying long-range part of the potential is evaluated through velocity jump processes. By design, the method adapts itself to minimize computational efforts to compute long-range forces while preserving accuracy. In the context of droplet simulations where the non-bonded forces are completely evaluated in direct space, the approach has been shown to have a similar accuracy as a BAOAB integrator with a 1 fs time step while being significantly faster with up to a 4-time acceleration. Significant computational speedup is also obtained in periodic boundary conditions with PME by combining BOUNCE with the RESPA method to evaluate the short-range part of the non-bonded forces and reciprocal-space contribution less often than the bonded terms: up to a factor 3.2 compared to BAOAB with a 1 fs time step. Furthermore, the use of BOUNCE always computationally outperforms the corresponding BAOAB-RESPA approaches while providing similar accuracy on all static and dynamical computed properties compared to the Verlet or BAOAB reference. Note that in this PBC context, the native basic BOUNCE integrator alone already provides enough direct-space acceleration to be computationally competitive with the traditional BAOAB-RESPA (0.5/2 fs) integrator. The purpose of the present work has essentially been to present the new algorithm, and for this reason, we did not yet apply all acceleration strategies that could be used in conjunction with BOUNCE. For example, as the BOUNCE algorithm strongly minimizes the computation of electrostatics and van der Waals terms in direct space, the choice of any direct summation technique or continuum solvation procedure will strongly benefit from this new integrator. The BOUNCE advantage can also be pushed further in periodic boundary conditions as one could choose to modify the Ewald parameter to increase the computational effort put in direct space compared to reciprocal space (at a fixed Ewald convergence) to benefit more from the BOUNCE efficiency gain as given in Table V. Overall, the computational gains obtained with BOUNCE depend on its implementation and on the proportion of the cost of the slowly varying gradient evaluation (which is the one reduced by the algorithm) compared to the rest (neighbor list routines for example). It is important to point out that the present

implementation is only prototypical and the BOUNCE algorithmic structure will provide more room for code optimization,⁴¹ whereas further evolutions of the way to treat the many-body reciprocal-space interactions are under investigation. Finally, this new technology offers new promises for speeding up simulations using polarizable force fields^{19,42–44} that include more advanced and therefore more computationally challenging electrostatics, induction, and van der Waals contributions that will not need to be computed at each time step anymore. Incoming works will address all these points in the near future.

SUPPLEMENTARY MATERIAL

See the Appendix in the [supplementary material](#) for details of the simulations and additional details discussed in the text.

ACKNOWLEDGMENTS

This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 810367), project EMC2. Computations have been performed at GENCI on the Occigen machine (CINES, Montpellier, France) on Grant No. A0070707671.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- L. Lagardère, L.-H. Jolly, F. Lipparini, F. Aviat, B. Stamm, Z. F. Jing, M. Harger, H. Torabifard, G. A. Cisneros, M. J. Schnieders, N. Gresh, Y. Maday, P. Y. Ren, J. W. Ponder, and J.-P. Piquemal, "Tinker-HP: A massively parallel molecular dynamics package for multiscale simulations of large complex systems with advanced point dipole polarizable force fields," *Chem. Sci.* **9**, 956–972 (2018).
- R. Salomon-Ferrer, D. A. Case, and R. C. Walker, "An overview of the Amber biomolecular simulation package," *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **3**(2), 198–210 (2013).
- M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, "GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers," *SoftwareX* **1–2**, 19–25 (2015).
- J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kalé, and K. Schulten, "Scalable molecular dynamics with NAMD," *J. Comput. Chem.* **26**(16), 1781–1802 (2005).
- S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," *J. Comput. Phys.* **117**(1), 1–19 (1995).
- A.-P. Hynninen and M. F. Crowley, "New faster CHARMM molecular dynamics engine," *J. Comput. Chem.* **35**(5), 406–413 (2014).
- C. Kobayashi, J. Jung, Y. Matsunaga, T. Mori, T. Ando, K. Tamura, M. Kamiya, and Y. Sugita, "GENESIS 1.1: A hybrid-parallel molecular dynamics simulator with enhanced sampling algorithms on multiple computational platforms," *J. Comput. Chem.* **38**(25), 2193–2206 (2017).
- D. E. Shaw, R. O. Dror, J. K. Salmon, J. P. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff, B. Batson, K. J. Bowers, E. Chow, M. P. Eastwood, D. J. Ierardi, J. L. Klepeis, J. S. Kuskin, R. H. Larson, K. Lindorff-Larsen, P. Maragakis, M. A. Moraes, S. Piana, Y. Shan, and B. Towles, "Millisecond-scale molecular dynamics simulations on anton," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC'09* (ACM, New York, NY, USA, 2009), pp. 39–1–39–11.
- B. Leimkuhler and C. Matthews, "Rational construction of stochastic numerical methods for molecular sampling," *Appl. Math. Res. eXpress* **2013**(1), 34–56.
- B. Leimkuhler and C. Matthews, "Robust and efficient configurational molecular sampling via Langevin dynamics," *J. Chem. Phys.* **138**(17), 174102 (2013).
- M. Tuckerman, B. J. Berne, and G. J. Martyna, "Reversible multiple time scale molecular dynamics," *J. Chem. Phys.* **97**(3), 1990–2001 (1992).
- J. Biesiadecki and R. D. Skeel, "Dangers of multiple time step methods," *J. Comput. Phys.* **109**(2), 318–328 (1993).
- Q. Ma, J. A. Izaguirre, and R. D. Skeel, "Verlet-I/R-RESPA/impulse is limited by nonlinear instabilities," *SIAM J. Sci. Comput.* **24**(6), 1951–1973 (2003).
- J. A. Morrone, T. E. Markland, M. Ceriotti, and B. J. Berne, "Efficient multiple time scale molecular dynamics: Using colored noise thermostats to stabilize resonances," *J. Chem. Phys.* **134**(1), 014103 (2011).
- B. Leimkuhler, D. T. Margul, and M. E. Tuckerman, "Stochastic, resonance-free multiple time-step algorithm for molecular dynamics with very large time steps," *Mol. Phys.* **111**(22–23), 3579–3594 (2013).
- D. T. Margul and M. E. Tuckerman, "A stochastic, resonance-free multiple time-step algorithm for polarizable models that permits very large time steps," *J. Chem. Theory Comput.* **12**(5), 2170–2180 (2016).
- A. Albaugh, M. E. Tuckerman, and T. Head-Gordon, "Combining iteration-free polarization with large time step stochastic-isokinetic integration," *J. Chem. Theory Comput.* **15**(4), 2195–2205 (2019).
- K. A. Feenstra, B. Hess, and H. J. C. Berendsen, "Improving efficiency of large time-scale molecular dynamics simulations of hydrogen-rich systems," *J. Comput. Chem.* **20**(8), 786–798 (1999).
- L. Lagardère, F. Aviat, and J.-P. Piquemal, "Pushing the limits of multiple-time-step strategies for polarizable point dipole molecular dynamics," *J. Phys. Chem. Lett.* **10**(10), 2593–2599 (2019).
- M. E. Tuckerman, B. J. Berne, and A. Rossi, "Molecular dynamics algorithm for multiple time scales: Systems with disparate masses," *J. Chem. Phys.* **94**, 1465 (1991).
- D. A. Gibson and E. A. Carter, "Time-reversible multiple time scale *ab initio* molecular dynamics," *J. Phys. Chem.* **97** 13429–13434 (1993).
- E. A. J. F. Peters and G. de With, "Rejection-free Monte Carlo sampling for general potentials," *Phys. Rev. E* **85**, 026703 (2012).
- P. Monmarché, "Piecewise deterministic simulated annealing," *Lat. Am. J. Probab. Math. Stat.* **13**(1), 357–398 (2016).
- P. Vanetti, A. Bouchard-Côté, G. Deligiannidis, and A. Doucet, "Piecewise deterministic Markov chain Monte Carlo," [arXiv:1707.05296](#) (2017).
- A. Durmus, A. Guillin, and P. Monmarché, "Piecewise deterministic Markov processes and their invariant measure," [arXiv:1807.05421](#) (2018).
- P. Monmarché, "Kinetic walks for sampling," *Lat. Am. J. Probab. Math. Stat.* **17**, 491 (2020).
- B. Leimkuhler, C. Matthews, and G. Stoltz, "The computation of averages from equilibrium and nonequilibrium Langevin molecular dynamics," *IMA J. Numer. Anal.* **36**(1), 13–79 (2016).
- A. Durmus, A. Guillin, and P. Monmarché, "Geometric ergodicity of the bouncy particle sampler," [arXiv:1807.05401](#) (2018).
- P. Monmarché, "Elementary coupling approach for non-linear perturbation of Markov processes with mean-field jump mechanisms and related problems," [arXiv:1809.10953](#) (2018).
- N. Bou-Rabee, "Time integrators for molecular dynamics," *Entropy* **16**, 138–162 (2014).
- J. Wang, P. Cieplak, and P. A. Kollman, "How well does a restrained electrostatic potential (resp) model perform in calculating conformational energies of organic and biological molecules?," *J. Comput. Chem.* **21**(12), 1049–1074 (2000).
- J. Huang, S. Rauscher, G. Nawrocki, T. Ran, M. Feig, B. L. de Groot, H. Grubmüller, and A. D. MacKerell, Jr., "CHARMM36m: An improved force field for folded and intrinsically disordered proteins," *Nat. Methods* **14**(1), 71–73 (2017).

- ³³U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, "A smooth particle mesh Ewald method," *J. Chem. Phys.* **103**(19), 8577–8593 (1995).
- ³⁴R. Zhou, E. Harder, H. Xu, and B. J. Berne, "Efficient multiple time step method for use with Ewald and particle mesh Ewald for large biomolecular systems," *J. Chem. Phys.* **115**(5), 2348–2358 (2001).
- ³⁵W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, "Comparison of simple potential functions for simulating liquid water," *J. Chem. Phys.* **79**(2), 926–935 (1983).
- ³⁶H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma, "The missing term in effective pair potentials," *J. Phys. Chem.* **91**(24), 6269–6271 (1987).
- ³⁷See <https://github.com/TinkerTools/Tinker-HP> for Tinker-HP's github; accessed 01 February 2020.
- ³⁸See <http://tinker-hp.ip2ct.upmc.fr/> for Tinker-HP's website; accessed 01 February 2020.
- ³⁹J. A. Morrone, R. Zhou, and B. J. Berne, "Molecular dynamics with multiple time scales: How to avoid pitfalls," *J. Chem. Theory Comput.* **6**(6), 1798–1804 (2010).
- ⁴⁰G. Bussi, D. Donadio, and M. Parrinello, "Canonical sampling through velocity rescaling," *J. Chem. Phys.* **126**(1), 014101 (2007).
- ⁴¹L.-H. Jolly, A. Duran, L. Lagardère, J. W. Ponder, P. Ren, and J.-P. Piquemal, "Raising the performance of the Tinker-HP molecular modeling package [article v1.0]," *Living J. Comput. Mol. Sci.* **1**(2), 10409 (2019).
- ⁴²J. Melcr and J.-P. Piquemal, "Accurate biomolecular simulations account for electronic polarization," *Front. Mol. Biosci.* **6**, 143 (2019).
- ⁴³Y. Shi, P. Ren, M. Schnieders, and J.-P. Piquemal, "Polarizable force fields for biomolecular modeling," in *Reviews in Computational Chemistry* (John Wiley & Sons, Ltd., 2015), Chap. 2, pp. 51–86.
- ⁴⁴Z. Jing, C. Liu, S. Y. Cheng, R. Qi, B. D. Walker, J.-P. Piquemal, and P. Ren, "Polarizable force fields for biomolecular simulations: Recent advances and applications," *Annu. Rev. Biophys.* **48**(1), 371–394 (2019).