PHYSICAL REVIEW E **89**, 053314 (2014)

Molecular dynamics algorithm enforcing energy conservation for microcanonical simulations

Clara Salueña
Departament d'Enginyeria Mecànica, Escola Tècnica Superior d'Enginyeria Química (ETSEQ), Universitat Rovira i Virgili,
Avda. Dels Països Catalans 26, 43007 Tarragona, Spain

Josep Bonet Avalos[*]
Departament d'Enginyeria Química, Escola Técnica Superior d'Enginyeria Química (ETSEQ), Universitat Rovira i Virgili,
Avda. Dels Països Catalans 26, 43007 Tarragona, Spain

(Received 20 February 2014; published 21 May 2014)

A reversible algorithm [enforced energy conservation (EEC)] that enforces total energy conservation for microcanonical simulations is presented. The key point is the introduction of the discrete-gradient method to define the forces from the conservative potentials, instead of the direct use of the force field at the actual position of the particle. We have studied the performance and accuracy of the EEC in two cases, namely Lennard-Jones fluid and a simple electrolyte model. Truncated potentials that usually induce inaccuracies in energy conservation are used. In particular, the reaction field approach is used in the latter. The EEC is able to preserve energy conservation for a long time, and, in addition, it performs better than the Verlet algorithm for these kinds of simulations.

DOI: 10.1103/PhysRevE.89.053314 PACS number(s): 02.70.Ns, 05.10.−a, 05.20.Jj

I. INTRODUCTION

Molecular dynamics (MD) is a well-established simulation tool to study the thermodynamic and dynamic properties of many systems. The method is based on the solution of the equations of motion of the system at a given level of description. One typically finds MD approaches based on the solution of the classical Newton's equation of motion with intermolecular potentials, while others start from the quantum-mechanical equations of motion with a given electronic description, hybrid quantum-classical mechanical effects (QMM), among others [1–3,5,6].

Classical MD simulations [1–3,7] with interparticle potentials have been applied to many systems with interesting physical [8], chemical [9], and biochemical properties [10,11], and they have also been applied in the field of in engineering [12]. The natural thermodynamic ensemble for these kinds of MD simulations is the microcanonical ensemble (*NVE*) due to the conservative nature of the dynamics of the system, which to some extent complies with the requirement of constant total (internal) energy E if no external forces are acting on the system. An important point regarding the use of the MD method to obtain meaningful proper averages in the *NVE* ensemble is that the total energy E is kept under control within a given energy conservation tolerance ΔE. This is possible despite the fact that the numerical trajectories soon exponentially deviate from the, say, physical trajectories, due to truncation errors accumulated by the numerical procedure. If the numerical trajectories are somehow *plausible* trajectories for a physical system and the energy is kept reasonably constant, the system will still sample the microcanonical ensemble [1,3,4]. Nevertheless, the use of different thermostats and barostats [13–17] to fix the temperature and or the pressure of the system has become widespread under the implicit hypothesis of the weak coupling of the system, with the thermostat (barostat) device representing the heat (volume) reservoir [1,2,7]. In particular, the use of thermostats for canonical (*NVT*) simulations permits long-time runs without the stability problems due to the long-time breaking of the energy conservation of *NVE* simulations. Actually, very long-time simulations are more frequently found in biological systems whose typical time scales are far above the nanosecond, such as protein folding, where transitions of the ternary structure are of the order of 1 μs at least. Nowadays, protein folding simulations are being performed within this time range [18] and up to 1 ms [19].

The energy fluctuations and eventually the energy divergence at long time, in systems with only conservative forces in the *NVE* ensemble, are due to the fact that, in a sequence of M time steps,

$$\sum_{i=1}^{M}\sum_{\alpha=1}^{N}\vec{F}_i^{\alpha}\cdot\Delta\vec{r}_i^{\alpha}\neq-\Delta U. \tag{1}$$

Here N is the number of particles, $\vec{F}_i^{\alpha}$ is the total force on the α particle in the time step i, $\Delta\vec{r}_i^{\alpha}$ is the displacement of this particle in this particular time step, and U is the potential energy of the system. The use of Lennard-Jones forces, for instance, obtained from the derivative of the Lennard-Jones potential [1], makes the left-hand side of Eq. (1) different from the variation of the total potential energy of the system, ΔU in an *NVE* simulation. In other words, the variation of the potential energy, as obtained from the past and actual coordinates of the particles, is not exactly equal to the work done from the forces in this time range along the simulated discrete trajectory. As is well known, the use of symplectic reversible algorithms is a great advantage, since the simulated trajectories lie close to the *physical*, namely differential continuous trajectories of the system in phase space [1,3,4]. In addition, this kind of algorithm, for which the Verlet algorithm would be its paramount representative, provides the system with a much larger stability with regard to the energy catastrophe for long-time simulations. Indeed, simulations of many-particle

---

[*] josep.bonet@urv.cat

systems interacting under Lennard-Jones potentials show a very good energy conservation [1,2,20–24], with low-energy drift. However, when electrostatic interactions are present, numerical truncation errors induce an energy drift that makes long *NVE* simulations inviable for large systems with electrostatic interactions typical of biomolecular calculations, for example. Moreover, time reversibility plays a very important role in statistical mechanics. The detailed balance (DB) condition satisfied by the transition probabilities between states can be demonstrated from the time reversibility of the trajectories for Hamiltonian systems in the *NVE* ensemble [25]. If the DB condition is satisfied, it can be further proven that the system will naturally evolve toward the proper thermodynamic equilibrium characterized by N, V, and E [25].

Thus, although *NVT* or *NPT* simulations are successfully used in many fields, *NVE* simulations may still be interesting for different purposes, such as the simple direct evaluation of physicochemical properties of systems [26], in dynamic histogram reweighting calculations [27], evaluation of the density of states [28], replica exchange algorithms for configurational space sampling [29], as well as for nonequilibrium simulations [30,31], for example. Therefore, the development of algorithms for *NVE* simulations that could have control over the energy divergence at long time is a matter of conceptual interest but also of practical relevance.

In this article, we present a time-reversible algorithm with enforced energy conservation (EEC) in *NVE* simulations, based on the discrete gradient theory. Instead of discretizing Newton's equation of motion, we propose to use the Hamiltonian (H) as the fundamental quantity. This way of proceeding implies abandoning the use of the intermolecular force as the basic element for the MD simulation and replacing it by a discrete derivative of H along the also discrete trajectory, characterized by the positions and velocities of all particles at a given sequence of discrete times $0 \leqslant i\delta t \leqslant M\delta t$, where δt is the time step and M is the total number of time steps. The prescription is made to recover the usual Newtonian form of the force from such a *discrete* force used in the algorithm when $\delta t \to 0$. In exchange, Eq. (1) becomes an identity (to a given controlled precision) independent of the size of the time step δt. Of course, since physical time is continuous, the physical meaning of the simulation depends on the size of the time step, but the adequate size of the time step is not determined by the stability related to energy conservation. Finally, we have to mention that this approach has been used in problems related to classical mechanics [32–36], which are rooted in an old seminal paper by Courant [37]. Since classical MD with interparticle potentials is formally identical, our method should be considered as an approach within this general scheme based on the so-called *discrete gradients* discussed in these references.

The article is organized as follows. In Sec. II we describe the basic equations underlying the integration scheme, and we discuss its properties and structure in light of existing MD algorithms and methods. In Sec. III we apply the method to two relevant cases. First, we use the Lennard-Jones potential as a benchmark for typical thermodynamic and transport properties, comparing the results obtained with our method and with data found in the literature. Second, we study the properties of the model for electrolytes within the framework of the reaction-field method, which represents a quite harsh approximation to the electrostatic interactions prone to cause important energy drifts in standard *NVE* simulations. Finally, in Sec. IV we discuss the results found and the conclusions that can be drawn from our work.

II. BASIC EQUATIONS

The model is based on the existence of a Hamiltonian,

$$H = \sum_{\alpha} \frac{1}{2m_\alpha} \vec{p}^{\alpha} \cdot \vec{p}^{\alpha} + \frac{1}{2} \sum_{\alpha} \sum_{\beta \neq \alpha} V^{\alpha\beta}, \tag{2}$$

where $\vec{p}^{\alpha} = m_\alpha \vec{v}^{\alpha}$ is the momentum of the particle α, m_α is its mass, $\vec{v}^{\alpha}$ is its velocity, and $V^{\alpha\beta}$ is the pairwise potential between particles α and β, depending on their positions $\vec{r}^{\alpha}$ and $\vec{r}^{\beta}$ only through $|\vec{r}^{\beta} - \vec{r}^{\alpha}|$. Many-body and external potentials are discussed later on in this section. In the following, greek indices label particles while latin indices indicate time steps. The proposed equations of motion read [2,38]

$$\vec{r}^{\alpha}_{i+1} = \vec{r}^{\alpha}_{i} + \frac{1}{2}\left(\vec{v}^{\alpha}_{i+1} + \vec{v}^{\alpha}_{i}\right)\delta t, \tag{3}$$

$$\vec{v}^{\alpha}_{i+1} = \vec{v}^{\alpha}_{i} + \frac{1}{m_\alpha}\vec{F}^{\alpha}\delta t = \vec{v}^{\alpha}_{i} + \frac{1}{m_\alpha}\sum_{\beta\neq\alpha}\vec{F}^{\alpha\beta}\delta t. \tag{4}$$

The requirement that H is invariant under the transformations introduced by Eqs. (3) and (4), if no external forces are applied to the system, implies that the force in Eq. (4) cannot be simply the force field $\vec{F}^{\alpha\beta} = -\hat{r}^{\alpha\beta}\partial V^{\alpha\beta}/\partial r^{\alpha\beta}$ at a given point, where $\hat{r}^{\alpha\beta} \equiv \vec{r}^{\alpha\beta}/|\vec{r}^{\alpha\beta}|$. Enforcing energy conservation implies that the usual description of the force in classical MD simulations, treated as a point property defined from the local gradient of the potential at a given position, should be changed by an expression that takes into account the variation of the potential energy in a discrete displacement of the whole system in a δt. We propose that the force is then given by

$$\begin{aligned}\vec{F}^{\alpha\beta} &= -\left(\vec{r}^{\alpha\beta}_{i+1} + \vec{r}^{\alpha\beta}_{i}\right)\frac{V^{\alpha\beta}_{i+1} - V^{\alpha\beta}_{i}}{\left(\vec{r}^{\alpha\beta\,2}_{i+1} - \vec{r}^{\alpha\beta\,2}_{i}\right)} \\ &= -\left(\vec{r}^{\alpha\beta}_{i+1} + \vec{r}^{\alpha\beta}_{i}\right)\frac{V^{\alpha\beta}_{i+1} - V^{\alpha\beta}_{i}}{\left[\left(\vec{r}^{\alpha\beta}_{i+1} + \vec{r}^{\alpha\beta}_{i}\right)\cdot\left(\vec{r}^{\alpha\beta}_{i+1} - \vec{r}^{\alpha\beta}_{i}\right)\right]} \equiv \vec{F}^{\alpha\beta}_{i+1,i},\end{aligned} \tag{5}$$

where $\vec{r}^{\alpha\beta}_{i} \equiv \vec{r}^{\beta}_{i} - \vec{r}^{\alpha}_{i}$. Attention should be paid to this sign convention used throughout the paper. These expressions for the discrete gradient of two-body central forces were in fact proposed already in 1974 by LaBudde and Greenspan [32].

This expression for the force is only valid for pairwise additive central forces. Its actual form is dictated by the fact that in the limit $\delta t \to 0$, Eq. (5) yields the usual expression for pairwise central forces, namely $\vec{F}^{\alpha\beta} \to -\hat{r}^{\alpha\beta}\partial V^{\alpha\beta}/\partial r^{\alpha\beta}$. Furthermore, its evaluation only depends on the positions of the particles along the discrete trajectory. The expressions for the force and the equations of motion are symmetric under time reversal in the generalized sense that they are invariant under the exchange of the initial and final points in a time step, together with the change of sign of the velocities. In the last equality of Eq. (5), we have stressed explicitly that the force pertains not to a point but to an interval $i+1,i$. Finally, from Eqs. (3) and (4) together with the definition of the force, Eq. (5), it is straightforward to verify that $H_{i+1} - H_i = 0$ in a time step. Therefore, the time reversibility and the energy-conservation properties ensure that the DB condition will be formally satisfied between two states on the $H = E$ hypersurface in phase space. The question about the *shadow Hamiltonian* [21] does not apply since H is, at least formally, exactly conserved in our algorithm. Finally, we have to stress that the presented algorithm is area-preserving only up to $O(\delta t^2)$, unlike the Verlet algorithm, which is so independently of δt. Symplecticity and energy conservation cannot be simultaneously satisfied in numerical integrators in general. For a more detailed discussion, see [23,24].

The pairwise additivity of the forces notably simplifies the expression of the discrete gradient, since the Hamiltonian can be effectively split into different independent contributions that are individually used to calculate the discrete gradient. Clearly, if many-body forces were present in the system, such a splitting would be impossible and a different form of the discrete gradient of the Hamiltonian should be used instead. For a general analysis of a dynamical system with a Hamiltonian in the framework of the discrete gradient method, see Ref. [35].

To end this section, we briefly discuss a few additional properties that can be of formal interest. First, Eqs. (3) and (4) can be cast under the form of the usual Verlet algorithm, with a redefined expression for the force. Effectively, considering Eq. (3) for the step $i+1$ and subtracting the same equation for the step i, using Eq. (4) to eliminate the velocity difference in the time step, one arrives at the expression

$$\vec{r}^{\,\alpha}_{i+1} = 2\vec{r}^{\,\alpha}_{i} - \vec{r}^{\,\alpha}_{i-1} + \frac{1}{2m_\alpha}\sum_{\beta\neq\alpha}\left(\vec{F}^{\alpha\beta}_{i+1,i} + \vec{F}^{\alpha\beta}_{i,i-1}\right)\delta t^2. \qquad (6)$$

As compared with the usual Verlet algorithm, our formulation replaces the evaluation of the force at i by the mean of the forces in the intervals $i+1,i$ and $i,i-1$, which implicitly introduces information on the three points considered in the algorithm, due to its implicit nature.

Secondly, if there are conservative external forces related to the particle's position, the discrete gradient can be constructed along the same lines as for the pairwise additive forces. Let us assume that a potential $V^{\text{ext}}(\vec{r})$ exists and that we can calculate the gradient of this potential, $\vec{g}(\vec{r}) \equiv \partial V/\partial \vec{r}$. The proposed discrete gradient then reads

$$\vec{F}^{\text{ext}\,\alpha}_{i+1,1} = -\left[\vec{g}\left(\vec{r}^{\,\alpha}_{i+1}\right) + \vec{g}\left(\vec{r}^{\,\alpha}_{i}\right)\right]\frac{V^{\text{ext}}\left(\vec{r}^{\,\alpha}_{i+1}\right) - V^{\text{ext}}\left(\vec{r}^{\,\alpha}_{i}\right)}{\left[\vec{g}\left(\vec{r}^{\,\alpha}_{i+1}\right) + \vec{g}\left(\vec{r}^{\,\alpha}_{i}\right)\right]\cdot\left(\vec{r}^{\,\alpha}_{i+1} - \vec{r}^{\,\alpha}_{i}\right)}. \qquad (7)$$

With this prescription, $\vec{F}^{\text{ext}\,\alpha}_{i+1,1} \cdot (\vec{r}^{\,\alpha}_{i+1} - \vec{r}^{\,\alpha}_{i}) = -[V^{\text{ext}}(\vec{r}^{\,\alpha}_{i+1}) - V^{\text{ext}}(\vec{r}^{\,\alpha}_{i})]$ is identically satisfied. Moreover, the limit $\delta t \to 0$ reduces the expression to the usual form for the force derived from a external potential $V^{\text{ext}}(\vec{r})$, namely $\vec{F}^{\text{ext}\,\alpha} = -\vec{g}(\vec{r}^{\,\alpha})$. This form for the discrete gradient [32] used throughout the paper relies on the identification of the independent coordinates for the given contribution of the potential.

Thirdly and finally, if many-body forces are present, an additive consideration of the variation of the energy through the independent discrete variation of the particle coordinate or the interparticle coordinate, independently of the rest of the system, is no longer possible. The linearity of the differential variation of the coordinates that permits the evaluation of the forces in the mechanics of differentiable trajectories does not apply when discrete variations are to be considered. Therefore, a different strategy has to be introduced to construct many-body forces that would satisfy the exact energy conservation in a time step. Following [35], let us assume that the system evolves according to a many-body potential $V(\{\vec{r}^{\,\alpha}\})$, where $\{\vec{r}^{\,\alpha}\}$ stands for the complete set of particles' coordinates $(\vec{r}^{\,1},\vec{r}^{\,2},\ldots,\vec{r}^{\,\alpha},\ldots,\vec{r}^{\,N})$, and that we can define a gradient field in the usual way, $\vec{g}^{\,\beta}(\{\vec{r}^{\,\alpha}\}) = \partial V(\{\vec{r}^{\,\alpha}\})/\partial\vec{r}^{\,\beta}$. Then, one can propose a force based on the so-called *averaged vector field* [39],

$$\vec{F}^{\beta}_{i+1,i} = -\int_0^1 d\xi\, \vec{g}^{\,\beta}\left[\left\{\vec{r}^{\,\alpha}_{i+1}\xi + \vec{r}^{\,\alpha}_{i}(1-\xi)\right\}\right]. \qquad (8)$$

Thus, the work done by the forces in a time step gives the change in V, since for each coordinate it can be written that $d\vec{r}^{\,\beta} = d[\vec{r}^{\,\beta}_{i+1}\xi + \vec{r}^{\,\beta}_{i}(1-\xi)] = (\vec{r}^{\,\beta}_{i+1} - \vec{r}^{\,\beta}_{i})d\xi$. As we have already mentioned, the choice of the discrete gradient is not unique. We can find, for instance, the coordinate increment method [40] among other possibilities described in the literature [39].

Hence, a reversible discrete algorithm with, at least, formally exact conservation of the energy can be constructed for a general many-body problem.

III. STUDY OF DIFFERENT CASES

In this section, we will address two cases of interest to illustrate the capabilities of the proposed energy-conserving algorithm. The selection includes the Lennard-Jones fluid with a study of its more prominent equilibrium and dynamic properties, as well as the electrolyte model with a reaction field.

A. Computational details

The algorithm can be implemented into a molecular dynamics code following the usual procedures. The implicit integration requires that an outer loop performs the time integration from t to $t+\delta t$, whereas an inner loop refines the solution at time $t+\delta t$ until convergence is achieved within the desired tolerance. Here an ordinary iteration procedure has been used and a tolerance has been defined that uses the criterion $|(H_{i+1} - H_i)/H_i| <$ tol, with H_i, H_{i+1} being the total energy of the system of N particles at times t and $t+\delta t$, respectively. For $N = 500$, tolerances in the range of 10^{-8}–10^{-6} have been used, depending on the system density: the more dense systems require smaller time steps and consequently smaller tolerances tuned to keep under control smaller particle displacements. Together with this control of the tolerance, we have introduced a global criterion of convergence, $|H_{i+1} - H_0| < |H_i - H_0|$, which takes into account the initial value of the energy and acts as a restoring mechanism that controls global energy deviations due to truncation and roundoff errors. The combination of both criteria succeeds in keeping the energy constant within the desired tolerance. A proper selection of the tolerance must consider essentially the number of particles in the system and the value of the time step δt, taking into account the machine precision, which was double precision in our calculations.

The computer domain satisfies periodic boundary conditions, and particle interactions are treated according to the minimum image convention. In the present implementation of the algorithm, positions are updated first following Eq. (3). Then the force is computed using Eq. (5) and finally velocities are updated by means of Eq. (4). The iterative procedure makes intensive use of neighbor lists, which therefore become compulsory even for small systems. Special care must be taken during the iteration not to lose track of a particle being reintroduced from any of the (periodic) boundaries, which would otherwise give a wrong contribution to the force via the incorrect calculation of $\vec{r}^{\alpha\beta}$ in Eq. (5). This means that in practice both $V_i^{\alpha\beta}$ and the estimate $V_{i+1}^{\alpha\beta}$ must be computed at every iteration step, after the selection of the appropriate particle images.

We have set to five the maximum number of iterations. Occasionally the procedure may require more than five iterations per time step, and in this case the policy is reducing the time step by one-half, until convergence is achieved or a minimum $\delta t_{\min}$ is reached. If the initial value of the time step is set correctly, the procedure is robust and quickly converges. With the above ingredients, the energy is conserved within the selected tolerance.

Finally, a comment should be made with respect to the truncation of the different potentials that we will analyze along the article, which is related to the problem of a collision with a hard wall. This can be illustrated with an example. In the course of a simulation, a particle with total energy $E = E_1 \lesssim 0$, bound to a truncated Lennard-Jones or reaction field potential, will eventually find the steep barrier that prevents it from accessing the region $E \geqslant 0$. At this moment, the particle should be rebounded according to the conservation of energy and momentum. The important fact is that the EEC algorithm as presented above does not solve the instantaneous collision. The converged solution found after iteration is the initial or precollisional state, not the postcollisional state. Therefore, one must solve this contingency either by regularizing the potential at r_c so that it is always a state with the suitable potential energy to be found after iteration, or by detecting the instantaneous collision and rebounding the particle accordingly. Here we have chosen a regularization of the potential. For the case of the truncated Lennard-Jones potential, the limit $V_{\mathrm{LJ}}(r_c^-)$ is regularized by applying an exponentially decaying function of the form

$$V_{\mathrm{LJ}}(r) = V_{\mathrm{LJ}}(r)\left(1 - e^{-\frac{r_c - r}{d}}\right) \quad \text{for} \quad r \leqslant r_c, \tag{9}$$

where r_c is the cutoff distance and d is the regularizing width $d \ll r_c$. In the case of the reaction field potential, the regularizing function (RF) is given by

$$V_{\mathrm{RF}}(r) = V_{\mathrm{RF}}(r)\frac{1}{2}\left[1 - \tanh\left(\frac{r_c - r}{d}\right)\right] \tag{10}$$

for r smaller than the Verlet radius, in this case.

B. The Lennard-Jones fluid

The Lennard-Jones fluid is used here as a benchmark to validate our code, which implements the enforcing EEC, and is also useful to compare its predictions with those provided by simulations with the classical Verlet algorithm. Let us consider the Lennard-Jones potential [1,2],

$$V_{\mathrm{LJ}}(r) = 4\epsilon[(\sigma/r)^{12} - (\sigma/r)^6], \tag{11}$$

with the values $\epsilon = 1$ and $\sigma = 1$ in the usual dimensionless units. In the following, the potential is truncated at a distance r_c. Data on the average potential energy per particle, the pressure, and the diffusion coefficient for a reduced temperature $T^* = 0.85$ and densities ρ^* between 0.776 and 0.9 of a system consisting of $N = 500$ particles for a simply cut potential in the *NVE*ensemble are found in [41]. The cutoff distance is set to $r_c = 3\sigma$ and the value of the time step is $\delta t = 0.005$ as in this reference. The system was equilibrated at the required temperature using the Verlet scheme. Here we are interested in the results of the average potential energy per particle U^* and the pressure P^* of the system. The data obtained in our simulations along with the benchmark values are collected in Table I.

The values of Table I are estimates obtained after a linear regression from simulation results pertaining to six runs at temperatures in the close vicinity of the respective benchmark temperature. Following [41], the usual long-range correction was applied to the potential energy (per particle),

$$U_{\mathrm{LRC}} = 2\pi\rho \int_{r_c}^{\infty} dr\, r^2 V_{\mathrm{LJ}}(r), \tag{12}$$

and the pressure was computed through the virial integral,

$$P = k_B T\rho - \frac{2}{3}\pi\rho^2 \int_0^{r_c} dr\, r^3 g(r)\frac{dV_{\mathrm{LJ}}}{dr} + P_{\mathrm{LRC}} \tag{13}$$

with

$$P_{\mathrm{LRC}} = -\frac{2}{3}\pi\rho \int_{r_c}^{\infty} dr\, r^3 \frac{dV_{\mathrm{LJ}}}{dr}, \tag{14}$$

where $g(r)$ is the pair correlation function. The agreement between benchmark and simulation results is good except for the pressure at $\rho^* = 0.90$. This value has been checked through independent simulations using a different code that implements the Verlet scheme, the result being $P^* = 2.20 \pm 0.01$, which is much closer to the value of 2.26 obtained here

TABLE I. Results of the reduced potential energy per particle U^* and pressure P^* of the system, obtained from present simulations (EEC) with respect to benchmark values (B) [41]. The standard error affecting the last digit of the values shown is in parentheses. The potential is simply cut at the cutoff radius $r_c = 3$.

ρ^*	T^*	U^* (EEC)	U^* (B)	P^* (EEC)	P^* (B)
0.776	0.851	−5.510(1)	−5.517(1)	0.018(8)	0.030(6)
0.780	0.853	−5.534(1)	−5.533(1)	0.075(6)	0.072(5)
0.820	0.852	−5.793(2)	−5.803(1)	0.566(8)	0.573(5)
0.840	0.851	−5.913(1)	−5.909(1)	0.898(4)	0.910(6)
0.860	0.849	−6.030(1)	−6.027(1)	1.27(1)	1.282(5)
0.900	0.851	−6.235(0)	−6.234(1)	2.261(4)	2.544(6)

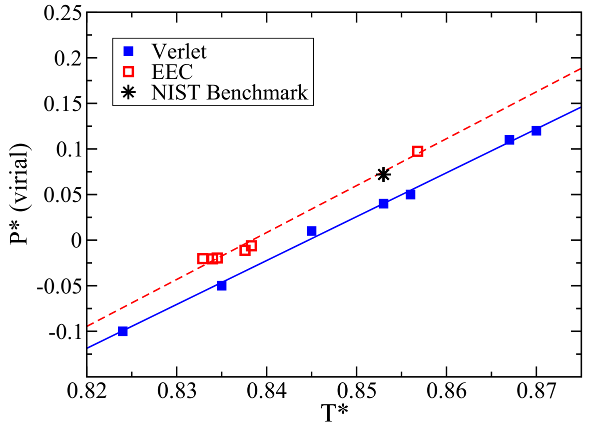

FIG. 1. (Color online) Values of the pressure obtained from the EEC and the standard Verlet algorithm using the virial equation (13) for different simulation runs, in comparison with the NIST benchmark [41] for $\rho^* = 0.78$ and the temperature $T^* = 0.853$. The estimate of the pressure at the benchmark temperature is made via linear regression on the simulated data.

than to the benchmark. In Fig. 1, we display the simulation results obtained for the density $\rho^* = 0.78$ and the temperature $T^* = 0.853$, as typical results. In general, we observe that the application of the classical Verlet algorithm produces slightly but consistently lower values for the pressure than the EEC algorithm. In Fig. 2, we compare the results for the pressure as obtained with a standard Verlet algorithm, the EEC algorithm, and the benchmark data.

Several remarks are in order regarding the pressure calculation. Most importantly, due to the fact that the expression of the force Eq. (5) involves a discrete gradient, the EEC algorithm effectively produces an impulsive force. Therefore, the pressure computed from the time average of the diagonal components of the stress tensor Eq. (A11) *does include this contribution* originating from the jump of the potential interaction energy at the truncation at r_c. This feature is relevant in some applications, but particularly in the comparison between Monte Carlo simulations with truncated potentials and molecular dynamics simulations. The Verlet algorithm instead uses the differential expression of the force, which is effectively truncated at r_c and thus does not produce any impulsive force contribution reflecting the discontinuity of the potential. The benchmark values shown in Table I for the EEC do not contain this term either, due to the fact that the pressure is calculated from the virial expression, which only requires structural data. The spatial distribution function $g(r)$ is not significantly affected by the impulsive force when compared to the equivalent obtained from the Verlet algorithm. To understand this important difference, let us consider the intermediate density 0.780 and the results obtained for the average pressure from the diagonal elements of the stress tensor, summarized in Fig. 3. The difference between the EEC and Verlet values (open red and filled blue squares, respectively) is significant. However, correcting the Verlet values of the pressure (green triangles) using a virial expression for the impulsive force [1] $P^*_{\text{av}} + P^{*\text{imp}}$, where

$$P^{\text{imp}} = \tfrac{8}{3}\pi\rho^2 r_c^3[(\sigma/r_c)^{12} - (\sigma/r_c)^6], \qquad (15)$$

shows that the agreement between the average pressure from the EEC and the corrected pressure from the Verlet is indeed excellent.

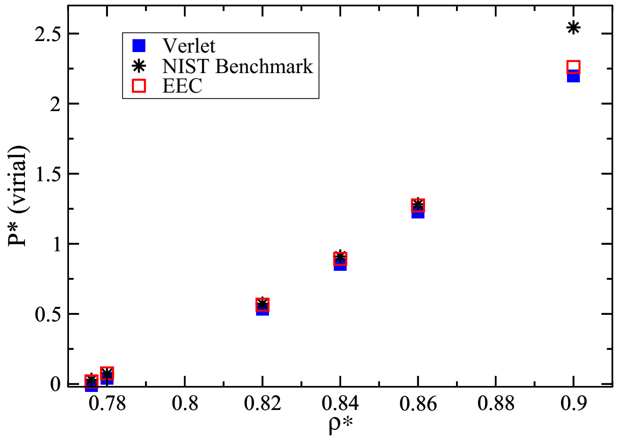

FIG. 2. (Color online) Values of the pressure as obtained from the EEC and the Verlet algorithm in comparison with the NIST benchmark [41].

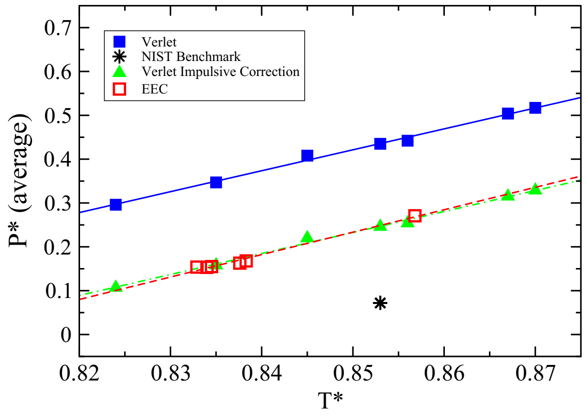

FIG. 3. (Color online) Values of the pressure obtained from the EEC and the standard Verlet algorithm using the diagonal components of the stress tensor, for $\rho^* = 0.78$ and the temperature $T^* = 0.853$. The green triangles correspond to the Verlet average values corrected with the impulsive contribution given in Eq. (15).

The previous analysis indicates that the thermodynamic properties obtained from the EEC algorithm are consistent with the properties derived from the Verlet algorithm. To determine the correctness of the predictions of the EEC model with regard to dynamic properties, we have calculated the viscosity η as well as the diffusion coefficient D for 8 thermodynamic conditions and compared the results with data in Ref. [42] (see Table II). In this case, we have used a cut-and-shifted Lennard-Jones potential with a cutoff distance $r_c^* = 4$, so that the EEC algorithm produces no impulsive force, as well as the Verlet algorithm. The results for the diffusion coefficient and the viscosity have been obtained using the corresponding Einstein relation (see Appendix B for details). Each EEC value in Table II is the average of three slopes from the respective linear regressions of the mean-square displacement, for D, and the correlation of the off-diagonal components of the

TABLE II. Comparison between the obtained diffusion coefficient D and viscosity η with the EEC algorithm and data from Tables III and IV of Ref. [42]. The conditions are the same as in this reference for each particular point.

ρ^*	T^*	D^* (EEC)	D^* (B)	η^* (EEC)	η^* (B)
0.4	1.3	0.3931(8)	0.392(2)	0.394(5)	0.39(4)
0.4	3	0.756(2)	0.752(6)	0.55(3)	0.54(6)
0.5	1	0.204(2)	0.216(3)	0.52(3)	0.53(7)
0.5	1.5	0.319(1)	0.318(3)	0.58(5)	0.56(4)
0.8	1	0.06501(4)	0.065(1)	2.0(2)	1.9(2)
0.8	2.1	0.1531(5)	0.148(1)	2.13(6)	1.9(5)
0.9	1	0.0365(1)	0.034(0)	4.0(1)	4.1(8)
1	1.8	0.050(2)	0.046(1)	5.1(4)	5.4(3)

stress tensor, for η, according to Appendix B. Again, we see very good agreement between the simulated EEC data and the reference data when similar parameters are used.

Finally, we analyze the different behavior of both algorithms with respect to the total energy conservation. As is discussed in the Introduction, the Verlet algorithm does not exactly conserve the energy, although the symplectic property introduces a global condition on the numerical trajectories. Intuitively, the conservation of the size of the volume element due to the symplecticity in the transformation means that close trajectories remain also close in time, despite the fact that numerical trajectories soon exponentially deviate from the physical trajectory. For a more detailed analysis, see Refs. [1,43]. The EEC algorithm suppresses these fluctuations up to the chosen tolerance. In Fig. 4, we have plotted the evolution of the total energy for both algorithms starting at exactly the same initial conditions. The difference in the average is due to the different way of calculating the kinetic energy in EEC with respect to Verlet. In the EEC case, the fluctuations in the total energy can be arbitrarily reduced by reducing the tolerance, given an adequate number of iterations.

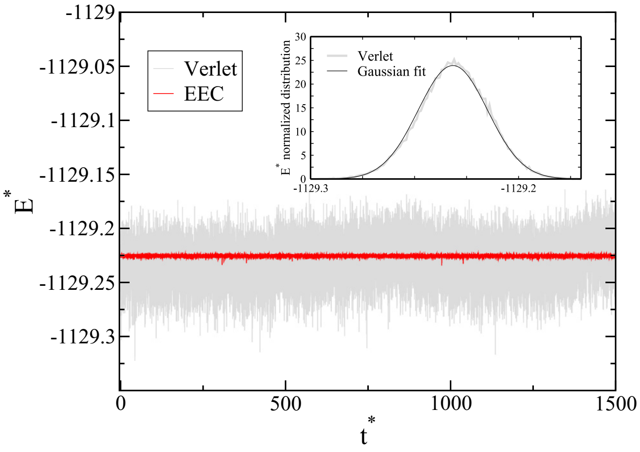

FIG. 4. (Color online) Total energy fluctuations for the EEC and Verlet algorithms for a Lennard-Jones fluid with 512 particles in a system of density $\rho^* = 0.5$ and temperature $T^*_{\mathrm{av}} = 0.99$. Both simulations started from the same equilibrated configuration. The average energy is $\langle E^* \rangle = -1129.2$ for the Verlet algorithm, while the EEC produces $\langle E^* \rangle = -1128.1$. The difference is due to the different ways of calculating the kinetic energy. In the graph, the EEC has been shifted onto the Verlet result for ease of the comparison.

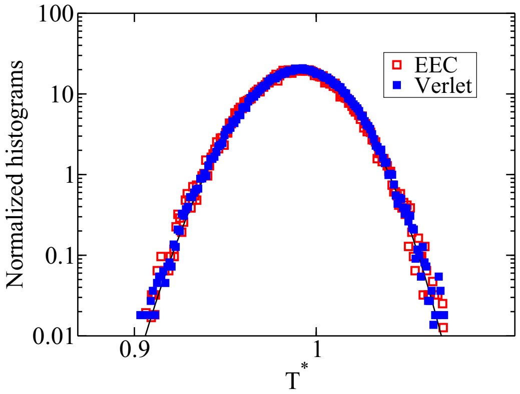

FIG. 5. (Color online) Normalized histograms of the temperature of the system of Fig. 4 for the EEC and Verlet algorithms. The continuous line is a Gaussian fit. The double logarithmic plot stresses the accuracy of the tails of the distribution.

The fluctuations in the total energy in the Verlet algorithm approximately follow a Gaussian distribution, as indicated in the inset of Fig. 4. Instead, the EEC energy fluctuations are controlled by the chosen value of the tolerance. However, and very remarkably, the temperature fluctuations follow the same Gaussian statistics in both cases, as shown in Fig. 5. Moreover, we have verified that the velocity distribution in the EEC calculations is perfectly Maxwellian.

C. Electrolyte model with a reaction field

As a second example to be studied, we have selected a simple system with electrostatic interactions. These long-range interactions are treated here through the reaction-field methodology, known to introduce harsh numerical errors in the total energy conservation. The methodology is widely described in the literature (see, for example, Refs. [44–46]). The electrostatic energy between charges in the system is pairwise additive and is given by

$$V^{\mathrm{RF}\,\alpha\beta} = \frac{1}{4\pi\varepsilon_0}\frac{q^\alpha q^\beta}{r^{\alpha\beta}}\left[1+\frac{\varepsilon_{\mathrm{RF}}-1}{2\varepsilon_{\mathrm{RF}}+1}\frac{r^{\alpha\beta\,3}}{r_c^{\mathrm{RF}\,3}}\right] \quad \text{for} \quad r^{\alpha\beta} \leqslant r_c^{\mathrm{RF}} \tag{16}$$

and $V^{\mathrm{RF}\,\alpha\beta} = 0$ otherwise. Here, r_c^{RF} is the reaction-field cutoff radius, and $\varepsilon_{\mathrm{RF}}$ is the reaction-field dielectric permittivity. We have set $\varepsilon_{\mathrm{RF}} = \infty$, which would correspond to a metallic material, which is suitable for a system with free charges. The force is then calculated using the prescription of Eq. (5) for central forces.

In the simulations, we have considered 1029 identical Lennard-Jones particles, from which 343 are neutral, simulating the solvent, 343 bear a net positive charge q^*, and the remaining 343 have an equal negative charge to make the overall system neutral. The reduced charge is defined according to

$$q^{*2} \equiv \frac{1}{4\pi\varepsilon_0}\frac{q^2}{\sigma\varepsilon}. \tag{17}$$

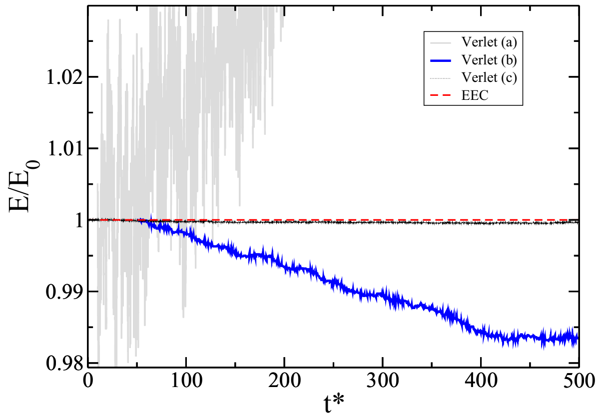

FIG. 6. (Color online) Comparison of the energy drift in EEC and Verlet algorithms, for the electrolyte model with a reaction field. The Verlet (a) line corresponds to the simulation with a simple step function at $r = r_c$ and $\delta t^* = 1.5 \times 10^{-3}$. Verlet (b) and (c) use a regularizing width of $d = 10^{-3} r_c$. They show that under the same regularization parameters, the drift is strongly sensitive to the value of the time step: (b) soon deviates with $\delta t^* = 1.5 \times 10^{-3}$ as compared with (c), for which $\delta t^* = 1 \times 10^{-3}$.

The thermodynamic conditions chosen are $T^* = 1.0$ and $\rho^* = 0.8$. The value of the reduced charge has been selected to avoid ion pairing in the system, i.e., for the Bjerum length to be smaller than the repulsive core of the particles, here estimated of the order of $r \simeq \sigma$ ($r^* \simeq 1$). Such a condition then reads $T^* \geqslant q^{*2}$, which is satisfied with a charge $q^* = 1$. The regularization chosen for the simulation is given in Eq. (10)

Figure 6 summarizes the different simulations regarding the electrolyte model. First, the use of Verlet algorithm with a nonregularized reaction-field potential produces not only large fluctuations but also a noticeable drift in the energy, independently of the size of the time step (light gray line). This result is an extreme situation that illustrates the great underlying difficulty of conducting microcanonical simulations with electrostatic potentials even for a relatively short time. The energy drift for this case is of the order of 0.5% per unit of time t^*. With the same time step $dt^* = 1.5 \times 10^{-3}$ as in the previous case, the introduction of a steep regularization ($d^* = 0.001\, r_c^*$), according to Eq. (10), causes the fluctuations to strongly decrease. The energy drift is also reduced (blue thick solid line), being for this case 0.2% per unit of time, which is still very high. In the third case, in addition to the regularization we have decreased the time step to $dt^* = 1.0 \times 10^{-3}$. Then, a relatively well-behaved dynamics is produced (black dotted line) within the shown time span. These results are compared to an EEC simulation with the same regularization width and time step $dt^* = 7.5 \times 10^{-3}$ (red dashed line). As it could not be otherwise, EEC keeps the total energy under the prescribed tolerance.

In general, the condition to be satisfied in order to have some control on the energy conservation using the Verlet algorithm is that $\bar{v}\delta t \ll d$ with $\bar{v}^2 \sim k_B T/m$, when regularization functions are introduced. Such a condition reduces the number of very energetic molecules at the tail of the velocity distribution that would overlook the steep change in the potential. The results obtained with regard to the energy conservation for the Verlet algorithm are therefore strongly dependent on the width of the regularization function and not on physical reasons for a given time step and temperature.

The following natural question to be asked is the relative performance of the EEC and the Verlet algorithms for the case in which the energy is conserved under comparable conditions, namely the systems represented by the black dotted [Verlet (c)] and red dashed lines in Fig. 6. Notice that we have chosen a relatively large time step for the EEC integration to stress the robustness of the procedure. Table III summarizes the results, obtained with a MATLAB code available to the reader as supplemental material [47].

We can see from the data collected in the table that the EEC algorithm performs better than the Verlet algorithm, with a total number of steps 60 867 < 100 000 for the former with respect to the latter. It is also worth noticing the consistency in the thermodynamic properties evaluated in the runs, which indicates a correct sampling of the microcanonical ensemble. Therefore, despite its higher complexity, the EEC algorithm is competitive for constant energy simulations in systems with long-range potentials. However, the difference in performance is not significantly larger for these relatively short-time runs. The key point in favor of the EEC algorithm comes from the long-time analysis of the Verlet integration. Effectively, a closer view of the black dotted [Verlet (c)] line over about 1000 reduced-time units indicates that there is a constant negative energy drift of the order of 1.0×10^{-5}% per unit of time, which would ruin a long-time microcanonical simulation of about 10^6 time units, for instance.

IV. DISCUSSION AND CONCLUSIONS

In this article, we have applied the concepts of the discrete gradient method to construct a time-reversible algorithm for its application to classical molecular dynamics simulations in the microcanonical ensemble. Enforcing energy conservation requires an implicit algorithm that introduces an inner iterative loop linked to a given precision in the energy conservation. The main idea of the method is to consider the energy as the central point, and to introduce a particular definition of the force from a discrete gradient instead of the straightforward gradient of the continuous potential. The concept of the

TABLE III. Comparison of the performance of Verlet and EEC algorithms under similar conditions. Both Verlet and EEC algorithms are written in MATLAB code [47], which makes runs last longer than with optimized codes written in a lower-level language. Here $\delta E/E_0$ stands for either the tolerance of the EEC or the range of the total energy fluctuations in the Verlet simulation.

	t^*	δt^*	$O(\delta E/E_0)$	Time steps	Iterations	Run time (s)	$U^*_{\rm av}$	$T^*_{\rm av}$	$P^*_{\rm av}$
Verlet	100	1.0×10^{-3}	1.0×10^{-4}	100 000		20 568	−6.61	1.03	0.46
EEC	100	7.5×10^{-3}	1.0×10^{-4}	13 334	60 867	18 423	−6.61	1.03	0.41

discrete gradient has been previously used in simulations of celestial dynamics, among others. We have here used a particularly suitable explicit form for central pairwise additive forces and external forces that are typically used in molecular dynamics simulations, which greatly simplifies the application of the discrete gradient method.

The EEC algorithm implements this idea for applications in molecular dynamics simulations, producing precise and reliable values of the thermodynamic and dynamic properties while keeping the total energy of the system constant except for small numerical fluctuations below the prescribed tolerance. These fluctuations are several orders of magnitude smaller than the natural total energy fluctuations produced by the Verlet algorithm with a comparable time step, although the long-time overall energy control of the latter is remarkable. However, in particularly steep potentials with long-range forces, we have shown that a small but persistent energy drift exists that cannot be neglected in long-time simulations. For these cases, the EEC algorithm is competitive and permits very long-time simulations with perfect control of the energy conservation.

With regard to the accuracy of the results, we have checked that the fluctuations in the total energy of the Verlet algorithm do not significantly affect either the thermodynamic properties or the dynamic quantities such as diffusion and viscosity coefficients. The EEC algorithm produces remarkably accurate results, including a correct accounting for the *impulsive* forces originating in steep potentials. The analysis of the electrolyte model, however, presented a challenge for the performance of the Verlet algorithm, which has traditionally been used in combination with different kinds of thermostats to avoid large energy fluctuations and energy drifts. These types of simulations require stability at long times, which is guaranteed by the EEC algorithm so that microcanonical simulations can be carried out for as long a time as desired. Furthermore, even at short times and compared to a similarly performing Verlet algorithm, the EEC is a competitive alternative. The application of the EEC to electrodynamic classical systems may open the possibility of an accurate study of the physical properties of these systems within the microcanonical ensemble.

ACKNOWLEDGMENTS

C.S. acknowledges the support from the Spanish Ministry of Economy and Competitiveness under project DPI2010-17212.

APPENDIX A: DERIVATION OF THE STRESS TENSOR FOR THE EEC

The theory is formulated as discrete, with the only condition that H is a constant of motion, together with the time-reversal invariance of the properties of the system. Thus, it is necessary to derive the observables related to dynamic properties in order to take into account the particular properties of the model.

The stress tensor is derived from the momentum transport in the long-wavelength limit $k \to 0$. The microscopic variable is the momentum density in Fourier space, i.e.,

$$\vec{j}_k = \sum_\alpha m_\alpha \vec{v}^\alpha \, e^{-i\vec{k}\cdot\vec{r}^\alpha}. \tag{A1}$$

We then calculate the variation of the momentum density in a δt. One has

$$\Delta \vec{j}_k = \sum_\alpha \left[m_\alpha \vec{v}^\alpha_{i+1} \, e^{-i\vec{k}\cdot\vec{r}^\alpha_{i+1}} - m_\alpha \vec{v}^\alpha_i \, e^{-i\vec{k}\cdot\vec{r}^\alpha_i} \right]. \tag{A2}$$

We then introduce new spatial variables that will allow us to gather terms of the same character. These are

$$\vec{r}^\alpha_{i+1} = \vec{R}^\alpha + \Delta\vec{r}^\alpha, \tag{A3}$$

$$\vec{r}^\alpha_i = \vec{R}^\alpha - \Delta\vec{r}^\alpha. \tag{A4}$$

Hence, $\vec{R}^\alpha = (\vec{r}^\alpha_{i+1} + \vec{r}^\alpha_i)/2$ and $\Delta\vec{r}^\alpha = (\vec{r}^\alpha_{i+1} - \vec{r}^\alpha_i)/2$. Then, expanding Eq. (A2) for $k\delta r \to 0$, one finds

$$\Delta \vec{j}_k = \sum_\alpha \left[m_\alpha \vec{v}^\alpha_{i+1}(1 - i\vec{k}\cdot\Delta\vec{r}^\alpha) - m_\alpha \vec{v}^\alpha_i (1 + i\vec{k}\cdot\Delta\vec{r}^\alpha) \right] \times e^{-i\vec{k}\cdot\vec{R}^\alpha}. \tag{A5}$$

Gathering terms on the right-hand side by its power of k, we find a first term that reads

$$\sum_\alpha \left[m_\alpha \vec{v}^\alpha_{i+1} - m_\alpha \vec{v}^\alpha_i \right] e^{-i\vec{k}\cdot\vec{R}^\alpha} = \sum_\alpha \vec{F}^\alpha \, e^{-i\vec{k}\cdot\vec{R}^\alpha} \delta t, \tag{A6}$$

where use has been made of Eq. (4). This term accounts for the force density. If we separate the external forces from the internal forces, the term will have a further natural expansion of the form

$$\sum_\alpha \vec{F}^\alpha \, e^{-i\vec{k}\cdot\vec{R}^\alpha} \delta t = \sum_\alpha \vec{F}^\alpha_{\text{ext}} \, e^{-i\vec{k}\cdot\vec{R}^\alpha} \delta t + \sum_\alpha \sum_{\beta\neq\alpha} \vec{F}^{\alpha\beta} \, e^{-i\vec{k}\cdot\vec{R}^\alpha} \delta t. \tag{A7}$$

The first term of the right-hand side of the preceding equation stands for the external *body* force density, which does not contribute to the stress. The second term, however, carries the so-called configurational contribution. Effectively, using the action reaction principle,

$$\sum_\alpha \sum_{\beta\neq\alpha} \vec{F}^{\alpha\beta} \, e^{-i\vec{k}\cdot\vec{R}^\alpha} \delta t = \sum_\alpha \sum_{\beta<\alpha} \vec{F}^{\alpha\beta} \left(e^{-i\vec{k}\cdot\vec{R}^\alpha} - e^{-i\vec{k}\cdot\vec{R}^\beta} \right) \delta t. \tag{A8}$$

Since these forces will be relatively short-range (smaller than the box size), we can write $\vec{R}^\beta = \vec{R}^\alpha + \vec{R}^{\alpha\beta}$ with $kR^{\alpha\beta} \to 0$. Then,

$$\begin{aligned} &\sum_\alpha \sum_{\beta<\alpha} \vec{F}^{\alpha\beta} \left(e^{-i\vec{k}\cdot\vec{R}^\alpha} - e^{-i\vec{k}\cdot\vec{R}^\beta} \right) \delta t \\ &\quad = i\vec{k} \cdot \sum_\alpha \sum_{\beta<\alpha} \vec{R}^{\alpha\beta} \vec{F}^{\alpha\beta} \, e^{-i\vec{k}\cdot\vec{R}^\alpha} \delta t. \end{aligned} \tag{A9}$$

The second contribution reads

$$-i\vec{k} \cdot \sum_\alpha \Delta\vec{r}^\alpha \left[m_\alpha \vec{v}^\alpha_{i+1} + m_\alpha \vec{v}^\alpha_i \right] e^{-i\vec{k}\cdot\vec{R}^\alpha}. \tag{A10}$$

Using Eq. (3), we can write $\Delta\vec{r}^{\alpha} = (\vec{v}^{\alpha}_{i+1} + \vec{v}^{\alpha}_{i})\delta t/4$. Then, the stress tensor is written as

$$\vec{\vec{\Pi}} \equiv \sum_{\alpha}\left(m_{\alpha}\frac{\left(\vec{v}^{\alpha}_{i+1} + \vec{v}^{\alpha}_{i}\right)}{2}\frac{\left(\vec{v}^{\alpha}_{i+1} + \vec{v}^{\alpha}_{i}\right)}{2} - \sum_{\beta<\alpha}\frac{\left(\vec{r}^{\alpha\beta}_{i+1} + \vec{r}^{\alpha\beta}_{i}\right)}{2}\vec{F}^{\alpha\beta}\right). \tag{A11}$$

APPENDIX B: EINSTEIN'S RELATION FOR THE SHEAR VISCOSITY η

Due to the long-time decay of the pressure tensor time correlation function, the computation of the viscosity through a Green-Kubo expression is very inefficient. However, the use of the Einstein relation [48] overcomes the effect of these long temporal tails in the Green-Kubo calculations, and we can obtain precise values of the viscosity with a reasonable computational effort. Consequently, the Einstein relation was employed as described by Smith *et al.* [48], but contrary to the original work, we use all the off-diagonal elements of the stress tensor (according to Ref. [49]) to improve convergence and statistics. Thus, the viscosity coefficient is given by the expression

$$\eta = \frac{1}{12}\frac{1}{V k_B T}\lim_{t\to\infty}\frac{d}{dt}\left[\sum_{x_i,x_j\neq x_i}\left\langle\Delta\Pi_{x_i x_j}(t)\right\rangle^2\right], \tag{B1}$$

where $x_i = x,y,z$ for $i = 1,2,3$ are the Cartesian coordinates, V is the volume, T is the temperature, and $\Delta\Pi_{x_i,x_j}(t)$ denotes the *displacement* of the elements of the stress tensor Π_{x_i,x_j}, according to

$$\Delta\Pi_{x_i,x_j}(t) = \int_0^t \Pi_{x_i,x_j}(\tau)d\tau. \tag{B2}$$

The microscopic expression for the elements of the pressure tensor $\Pi_{\alpha\beta}$ appearing in the integrand of Eq. (B2) is given by Eq. (A11). Therefore, due to the discreteness of the derivation of the stress tensor, the integral in Eq. (B2) can be replaced by the Riemann sum with δt instead of $d\tau$. The viscosity is obtained from the slope of Eq. (B1), always after some initial time, where the displacement is not a linear function of time.

---

[1] D. Frenkel and B. Smit, *Understanding Molecular Simulation*, 2nd ed. (Elsevier, Amsterdam, 2002).

[2] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon, Oxford, 1987).

[3] M. E. Tuckerman and G. J. Martyna, J. Phys. Chem. B **104**, 159 (2000).

[4] S. Toxvaerd, Phys. Rev. E **50**, 2271 (1994).

[5] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985).

[6] G. Galli and M. Parrinello, Comput. Simul. Mater. Sci. **3**, 283 (1991).

[7] D. C. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 2004).

[8] E. Bourasseau, M. Haboudou, A. Boutin, A. H. Fuchs, and Ph. Ungerer, J. Chem. Phys. **118**, 3020 (2003).

[9] J.-R. Li, Y. Mab, M. C. McCarthy, J. Sculley, J. Yub, H.-K. Jeong, P. B. Balbuena, and H.-C. Zhou, Coord. Chem. Rev. **255**, 1791 (2011).

[10] M. Karplus and J. A. McCammon, Nat. Struct. Biol. **9**, 646 (2002).

[11] M. Karplus and J. Kuriyan, Proc. Natl. Acad. Sci. (USA) **102**, 6679 (2005).

[12] A. Jabbarzadeh and R. I. Tanner, Rheol. Rev. 165 (2006).

[13] S. Nosé, Mol. Phys. **52**, 255 (1984).

[14] S. Nosé, Prog. Theor. Phys. Suppl. **103**, 1 (1991).

[15] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[16] W. G. Hoover, Phys. Rev. A **34**, 2499 (1986).

[17] H. C. Andersen, J. Chem. Phys. **72**, 2384 (1980).

[18] P. L. Freddolino, F. Liu, M. Gruebele, and K. Schulten, Biophys. J. **94**, L75 (2008).

[19] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, and W. Wriggers, Science **330**, 341 (2010).

[20] L. Verlet, Phys. Rev. **159**, 98 (1967).

[21] J. Gans and D. Shalloway, Phys. Rev. E **61**, 4587 (2000).

[22] R. D. Engle, R. D. Skeel, and M. Drees, J. Comput. Phys. **206**, 432 (2005).

[23] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems* (Chapman & Hall, London, 1994).

[24] E. Celledoni, V. Grimm, R. I. Mclachlan, D. I. Mclaren, D. O'neale, B. Owren, and G. R. W. Quispel, J. Comput. Phys. **231**, 6770 (2012).

[25] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry* (North-Holland, Amsterdam, 1990), Chap. V, Sec. 6.

[26] P. Mark and L. Nilsson, J. Phys. Chem. A **105**, 9954 (2001).

[27] C. Nieto-Draghi, J. Pérez-Pellitero, and J. B. Avalos, Phys. Rev. Lett. **95**, 040603 (2005).

[28] S. Davis, J. Phys.: Conf. Ser. **410**, 012161 (2013).

[29] P. Kar, W. Nadler, and U. H. E. Hansmann, Phys. Rev. E **80**, 056703 (2009).

[30] F. Muller-Plathe, J. Chem. Phys. **106**, 6082 (1997).

[31] C. Nieto-Draghi and J. Bonet Avalos, Mol. Phys. **101**, 2303 (2003).

[32] R. A. LaBudde and D. Greenspan, J. Comput. Phys. **15**, 134 (1974).

[33] O. Gonzalez, J. Non-lin. Sci. **6**, 449 (1996).

[34] R. I. McLachlan, G. R. W. Quispel, and N. Robidoux, R. Soc. London Philos. Trans. Ser. A **357**, 1021 (1999).

[35] M. Dahlby, B. Owren, and T. Yaguchi, J. Phys. A **44**, 305205 (2011).

[36] M. Dahlby and B. Owren, SIAM J. Sci. Comput. **33**, 2318 (2011).

[37] R. Courant, K. Friedrichs, and H. Lewy, Math. Ann. **100**, 32 (1928).

[38] A. Rahman, Phys. Rev. **136**, A405 (1964).

[39] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, 2nd ed., Springer Series in Computational Mathematics Vol. 31 (Springer-Verlag, Berlin, 2006).

[40] T. Itoh and K. Abe, J. Comput. Phys. **76**, 85 (1988).

[41] http://www.cstl.nist.gov/srs/LJ_PURE/md.htm.

[42] R. L. Rowley and M. M. Painter, Int. J. Thermophys. **18**, 1109 (1997).

[43] S. Reich, SIAM J. Numer. Anal. **36**, 1549 (1999).

[44] W. F. van Gunsteren, H. J. C. Berendsen, and J. A. C. Rullmann, Faraday Discuss. **66**, 58 (1978).

[45] M. Neumann, O. Steinhauser, and G. S. Pawley, Mol. Phys. **52**, 97 (1984).

[46] M. Neumann, Mol. Phys. **57**, 97 (1986).

[47] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.89.053314 for the implementation of the EEC algorithm in MATLAB.

[48] P. E. Smith and W. F. van Gunsteren, Chem. Phys. Lett. **215**, 315 (1993).

[49] D. K. Dysthe, A. H. Fuchs, and B. Rousseau, J. Chem. Phys. **110**, 4047 (1999).