An enhanced version of the heat exchange algorithm with excellent energy conservation properties.

P. Wirnsberger,[1, a)] D. Frenkel,[1] and C. Dellago[2]

[1)]*Department of Chemistry, University of Cambridge, Cambridge CB2 1EW, United Kingdom*

[2)]*Faculty of Physics, University of Vienna, 1090 Vienna, Austria*

(Dated: 29th October, 2015)

We propose a new algorithm for non-equilibrium molecular dynamics simulations of thermal gradients. The algorithm is an extension of the heat exchange algorithm developed by Hafskjold and co-workers [Mol. Phys. **80**, 1389 (1993); Mol. Phys. **81**, 251 (1994)], in which a certain amount of heat is added to one region and removed from another by rescaling velocities appropriately. Since the amount of added and removed heat is the same and the dynamics between velocity rescaling steps is Hamiltonian, the heat exchange algorithm is expected to conserve the energy. However, it has been reported previously that the original version of the heat exchange algorithm exhibits a pronounced drift in the total energy, the exact cause of which remained hitherto unclear. Here, we show that the energy drift is due to the truncation error arising from the operator splitting and suggest an additional coordinate integration step as a remedy. The new algorithm retains all the advantages of the original one whilst exhibiting excellent energy conservation as illustrated for a Lennard-Jones liquid and SPC/E water.

PACS numbers: 05.10.-a, 05.70.Ln, 02.60.Jh, 02.70.Ns
Keywords: non-equilibrium molecular dynamics, NEMD, heat exchange algorithm, HEX, thermal gradient

I. INTRODUCTION

Non-equilibrium molecular dynamics (NEMD) simulations allow us to study transport phenomena and determine transport coefficients[1]. In studies of heat conduction, an external field is applied to the system thereby driving it to a steady state. The nature of the coupling between the external field and the system differs between algorithms and determines whether a spatially homogeneous state[2], a temperature gradient[3–5] or a heat flux is imposed[6–9]. A suitable algorithm for a particular application depends on its ability to model the underlying physics correctly. If energy is supplied at a constant rate in an experiment, for example, a thermostat which imposes a heat flux would lend itself for the simulation. From a computational point of view, generating a flux might be preferable, because it is simpler to measure the temperature than the heat flux.

One way to generate a heat flux in computer simulations involves swapping kinetic energy between two subdomains of the simulation box[6,7]. In the heat exchange (HEX) algorithm developed by Ikeshoji and Hafskjold[6,10], a specific amount of heat is periodically removed from one subdomain or reservoir, and supplied to the other. These two regions thus act as a heat sink and source, respectively. The HEX method adjusts the non-translational kinetic energy by velocity rescaling while preserving the individual center of mass velocities of the two heat reservoirs. Other methods use different procedures to generate heat fluxes. In the reverse NEMD (RNEMD) method developed by Müller-Plathe[7], the heat transfer is established by continuously identifying hot and cold particles inside the reservoirs and exchanging their momenta. Extensions of the RNEMD method were proposed by Kuang and Gezelter[8,9], who replaced the momenta swaps by velocity rescaling moves. The velocity scaling and shearing (VSS) RNEMD method[9] allows for imposing a momentum flux in addition to the thermal flux. However, we note that in the absence of any momentum flux, the method is identical to the HEX algorithm. Although these methods are widely applicable, they all lack an attractive feature which is a formulation based on time continuous equations of motion. Knowing the equations of motion is advantageous, for example, if one is interested in studying system properties such as phase space compressibility or the development of accurate integration schemes.

Due to its simplicity, the HEX algorithm is an attractive choice for simulating a fluid in the absence of solid inhomogeneities. Since the same amount of energy is added and removed, one would expect the algorithm to conserve the total energy exactly. However, as pointed out in one of the original papers[10] and subsequent work[11], numerical implementations of the algorithm lead to a considerable energy drift over simulation time scales of a few nanoseconds. A change in total energy of several percent of the initial value was considered acceptable in past work. Nevertheless, the energy drift is a severe restriction limiting the accessible simulation time scales. Remedies to this problem either involve employing a smaller timestep or compensating the energy drift with an additional thermostat[12] which is undesirable, because such thermostats may affect the very temperature profile that one aims to study.

In this work, we identify the underlying cause of the energy loss and suggest a new algorithm to achieve improved energy conservation. The paper is organised as follows: In Sec. II, we summarise the HEX algorithm and its numerical implementation. We then show that the integration scheme leads to an unphysical energy drift. In order to understand the origin of this problem, we derive the equations of motion for continuous time in Sec. III. This allows us to express the integration scheme as a Trotter factorisation of the Liouville operator. In Sec. IV, we work out the leading-order error term of the employed operator splitting. Based on our analysis, we propose an enhanced algorithm in Sec. V and compare the results in Sec. VI.

[a)]Correspondence author. E-mail: pw359@cam.ac.uk

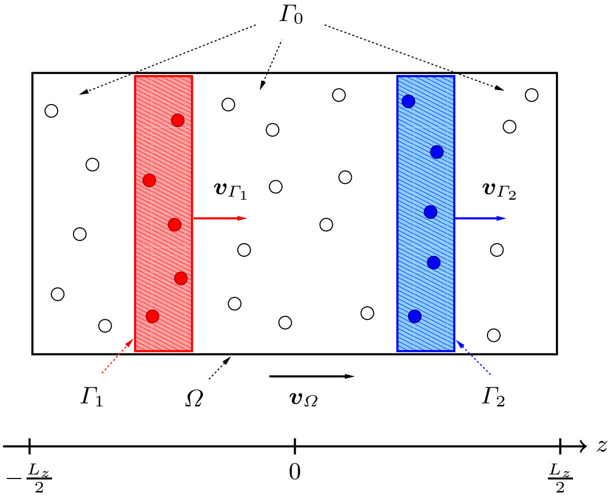

FIG. 1. Illustration of the simulation box, $\mathit{\Omega}$, with Hamiltonian regions, $\mathit{\Gamma}_0$, a hot region, $\mathit{\Gamma}_1$ (red), and a cold region, $\mathit{\Gamma}_2$ (blue). The centre of mass velocities of $\mathit{\Omega}$, $\mathit{\Gamma}_1$ and $\mathit{\Gamma}_2$ are $\boldsymbol{v}_{\mathit{\Omega}}$, $\boldsymbol{v}_{\mathit{\Gamma}_1}$ and $\boldsymbol{v}_{\mathit{\Gamma}_2}$, respectively. Atoms are represented by red/blue circles, if they are located in the hot/cold region and by empty circles otherwise.

II. HEX ALGORITHM

The goal of the HEX algorithm is to impose a constant heat flux onto the system. This is accomplished by adding heat $\Delta Q_{\mathit{\Gamma}_k}$ at each timestep to $N_{\mathit{\Gamma}}$ pair-wise disjoint subdomains $\mathit{\Gamma}_k$, of the simulation box $\mathit{\Omega}$ (Fig. 1). Heat is subtracted if $\Delta Q_{\mathit{\Gamma}_k}$ is negative. We label those parts of the simulation box which are not thermostatted with $\mathit{\Gamma}_0$. The box contains N atoms each labelled with a unique index. If there is no net energy flux into the simulation box as we assume here, i.e. $\sum_k \Delta Q_{\mathit{\Gamma}_k} = 0$, the system will approach a steady state in which heat fluxes are established between the subdomains. The position and velocity vectors of atom i are $\boldsymbol{r}_i$ and $\boldsymbol{v}_i$, respectively. Furthermore, we use $\boldsymbol{v}_{\mathit{\Gamma}_k}$ and $\boldsymbol{v}_{\mathit{\Omega}}$ to denote the centre of mass velocities of the regions $\mathit{\Gamma}_k$ and the box $\mathit{\Omega}$, respectively.

A. Energy supply

Energy is added or removed by rescaling the velocities of all particles contained in region $\mathit{\Gamma}_k$ by the same factor ξ_k and shifting them by a constant. The value of ξ_k is chosen such that the non-translational kinetic energy of that region,

$$\mathcal{K}_{\mathit{\Gamma}_k} = \sum_{i \in \gamma_k} \frac{m_i v_i^2}{2} - \frac{m_{\mathit{\Gamma}_k} v_{\mathit{\Gamma}_k}^2}{2}, \tag{1}$$

changes by $\Delta Q_{\mathit{\Gamma}_k}$ leaving $\boldsymbol{v}_{\mathit{\Gamma}_k}$ unchanged, where $m_{\mathit{\Gamma}_k}$ is the total mass contained in $\mathit{\Gamma}_k$. The time-dependent index set γ_k comprises all particles which are located in $\mathit{\Gamma}_k$. Particles outside any thermostatted region are not affected by this procedure. For the individual region $\mathit{\Gamma}_k$, the velocity update can be formulated as[6,13]

$$\boldsymbol{v}_i \mapsto \bar{\boldsymbol{v}}_i = \xi_k \boldsymbol{v}_i + \big(1 - \xi_k\big) \boldsymbol{v}_{\mathit{\Gamma}_k}, \tag{2}$$

where the rescaling factor is given by

$$\xi_k = \sqrt{1 + \frac{\Delta Q_{\mathit{\Gamma}_k}}{\mathcal{K}_{\mathit{\Gamma}_k}}}. \tag{3}$$

Here, updated quantities are denoted with an overbar. It can easily be verified that the update step given by Eq. (2) satisfies $\bar{\mathcal{K}}_{\mathit{\Gamma}_k} = \mathcal{K}_{\mathit{\Gamma}_k} + \Delta Q_{\mathit{\Gamma}_k}$ and $\bar{\boldsymbol{v}}_{\mathit{\Gamma}_k} = \boldsymbol{v}_{\mathit{\Gamma}_k}$. Since there is no net energy flux into the system according to our assumptions, this also implies that the total system energy E remains constant.

We note that the above formulation of the velocity update as presented by Aubry *et al.*[13] is simpler than the one which was originally proposed by Ikeshoji and Hafskjold[6]. In the latter case, ξ_k is a more complex function of the velocities, but it is easy to see that both formulations are equivalent.

B. Time integration

In order to keep track of the time evolution, it is convenient to introduce some additional notation. We label all quantities sampled at time $t = n\Delta t$ with a superscript n, where Δt is the timestep. In addition we define ξ_0 to be unity at all times and $k(\boldsymbol{r}_i)$ to be the index of the region in which particle i is located. The current state of the system is fully described by a $6N$-dimensional vector $\boldsymbol{x} = (\boldsymbol{r}, \boldsymbol{v})$ in phase space, where the vectors $\boldsymbol{r}$ and $\boldsymbol{v}$ contain all particle positions and velocities, respectively. The HEX algorithm for velocity Verlet[25] can then be formulated as

$$\bar{\boldsymbol{v}}_i^{n} = \xi^n_{k(\boldsymbol{r}_i)} \boldsymbol{v}_i^n + \Big(1 - \xi^n_{k(\boldsymbol{r}_i)}\Big) \boldsymbol{v}^n_{\mathit{\Gamma}_{k(\boldsymbol{r}_i)}}, \tag{4a}$$

$$\bar{\boldsymbol{v}}_i^{n+\frac{1}{2}} = \bar{\boldsymbol{v}}_i^n + \frac{\Delta t}{2 m_i} \boldsymbol{f}_i^n, \tag{4b}$$

$$\boldsymbol{r}_i^{n+1} = \boldsymbol{r}_i^n + \Delta t \, \bar{\boldsymbol{v}}_i^{n+\frac{1}{2}}, \tag{4c}$$

$$\boldsymbol{f}_i^{n+1} = -\nabla_{\boldsymbol{r}_i} U(\boldsymbol{r}) \,|_{\boldsymbol{r} = \boldsymbol{r}^{n+1}}, \tag{4d}$$

$$\bar{\boldsymbol{v}}_i^{n+1} = \bar{\boldsymbol{v}}_i^{n+\frac{1}{2}} + \frac{\Delta t}{2 m_i} \boldsymbol{f}_i^{n+1}, \tag{4e}$$

$$\boldsymbol{v}_i^{n+1} = \bar{\xi}^{n+1}_{k(\boldsymbol{r}_i)} \bar{\boldsymbol{v}}_i^{n+1} + \Big(1 - \bar{\xi}^{n+1}_{k(\boldsymbol{r}_i)}\Big) \bar{\boldsymbol{v}}^{n+1}_{\mathit{\Gamma}_{k(\boldsymbol{r}_i)}}, \tag{4f}$$

where $U(\boldsymbol{r})$ is the potential energy and $\boldsymbol{f}_i$ the force acting on particle i. For the entire scheme to be symmetric, half the energy is supplied at the beginning of the timestep and the other half at the end. The scaling factors $\xi^n_{k(\boldsymbol{r}_i)}$ and $\bar{\xi}^{n+1}_{k(\boldsymbol{r}_i)}$ are evaluated using Eq. (3) at the states $\big(\boldsymbol{r}^n, \boldsymbol{v}^n\big)$ and $\big(\boldsymbol{r}^{n+1}, \bar{\boldsymbol{v}}^{n+1}\big)$, respectively.

We note that in the original work[6,10], the authors do not provide any details about when exactly the thermostatting step should happen. For comparison, we also tested an asymmetric version of the algorithm (HEX/a), where all the energy is supplied at the end of the timestep. In this case, the initial velocity update reduces to the identity operation.

C. Model system

We studied the energy conservation of the HEX algorithm for a Lennard-Jones (LJ) fluid using the simulation package

LAMMPS (version 9Dec14)[14]. The symmetric, pairwise LJ potential is given by[15]

$$u_{\mathrm{LJ}}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{5}$$

where ϵ is the depth of the potential and σ the effective atomic diameter. In order to rule out any effects due to simple spherical truncation of the potential, we employed a slightly modified potential which is given by[16]

$$u_{\mathrm{SF}}(r) = u_{\mathrm{LJ}}(r) - u_{\mathrm{LJ}}(r_{\mathrm{s}}) - (r - r_{\mathrm{s}})u'_{\mathrm{LJ}}(r_{\mathrm{s}}) \tag{6}$$

for $r \leq r_{\mathrm{s}}$ and zero otherwise, where r_{s} is the cutoff. From the functional form of Eq. (6) it is clear that $u_{\mathrm{SF}}(r)$ and $u'_{\mathrm{SF}}(r)$ are both continuous at the cutoff. We employed a value of $r_{\mathrm{s}}^* = 3$ for all simulations in this section. (Reduced quantities are labelled with an asterisk.)

D. Equilibration

The rectangular simulation box with dimensions $L_z^*/2 = L_x^* = L_y^* = 10.58$ comprised $N = 2000$ atoms resulting in a density of $\rho^* = 0.8444$. The thermodynamic conditions considered in this work are similar to those in Ref. 6. Starting from an initial lattice structure with zero linear momentum, the system was heated up to twice the target temperature of $T^* = 0.72$ and subsequently cooled down again at the same rate. The thermostatting during this initial period was achieved by velocity rescaling and the entire annealing process took 2.5×10^4 timesteps. The equations of motion were integrated with the velocity Verlet algorithm using a timestep of $\Delta t^* = 0.002$. We then increased the timestep to $\Delta t^* = 0.004$ and carried out a 2×10^5 timestep *NVT* simulation using a Nosé–Hoover thermostat[17,18] with a relaxation time of $\tau^* = 0.5$. During this run we computed the average system energy. Using the HEX algorithm, we then adjusted the energy of the last configuration and used it as input for another 2×10^5 timestep *NVE* equilibration run. This procedure allowed us to achieve an average equilibrium temperature of $T^* = (0.7200 \pm 0.0002)$. The error bar corresponds to one standard deviation of the error of the mean, the variance of which was estimated using block average analysis[15].

As a reference for the energy conservation in equilibrium, we carried out an additional set of *NVE* simulations at various timesteps. With the above protocol we matched the temperature of these runs to be close to the one inside the hot reservoir in the NEMD case. The average temperature was $T^* = (0.8400 \pm 0.0002)$.

E. Energy conservation during NEMD

The previously equilibrated structures were subjected to a temperature gradient along the z-axis using the HEX algorithm. Always starting from the same phase space point, we varied the timestep for a fixed energy flux $\mathcal{F}_{\Gamma_k} = \Delta Q_{\Gamma_k}/\Delta t$ into the reservoir. The two thermostatted regions are centred at the points $z = \pm L_z/4$ and have a width of 2 in reduced units (Fig. 1). During each timestep, the heat ΔQ is taken from Γ_2 and added to Γ_1 ($\Delta Q_{\Gamma_1} = -\Delta Q_{\Gamma_2} = \Delta Q > 0$). We waited for 100 reduced time units for any transient behaviour to disappear and to allow the system to reach a steady state. The production run of 5000 reduced time units started at $t^* = 0$.

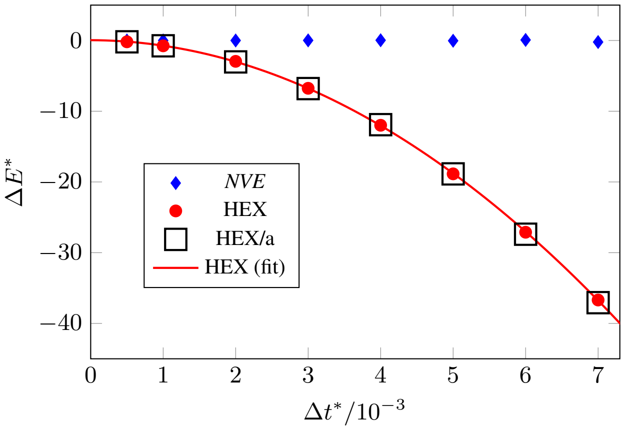

FIG. 2. Energy loss for LJ at the final time $t^* = 5000$ as a function of the timestep. Each point in the figure corresponds to a separate simulation. The equilibrium run (blue diamonds) is compared to the symmetric (red circles) and asymmetric (black squares) versions of the HEX algorithm, respectively. A quadratic fit (red, solid line) was carried out for the symmetric version.

In order to capture the spatial variation of the temperature, we divided the z-axis into N_{b} bins. We use the notation X_j for the evaluation of a quantity X over bin j and assign the value to the centre of the bin. The instantaneous kinetic temperature of bin j is then given by

$$T_j = \frac{2\mathcal{K}_j}{(N_j f - 3)k_{\mathrm{B}}}, \tag{7}$$

where $\mathcal{K}_j$ is the total non-translational kinetic energy of the bin, N_j the number of atoms contained in the bin and k_{B} Boltzmann's constant. The quantity f is the number of degrees of freedom per atom ($f_{\mathrm{LJ}} = 3$ and $f_{\mathrm{SPC/E}} = 2$). We subtracted three degrees of freedom to account for the centre of mass velocity of the bin. In the stationary state, the heat flux between the reservoirs in Fig. 1 is given by

$$J_{Q,z} = \frac{\Delta Q}{2\Delta t L_x L_y} = \frac{\mathcal{F}}{2L_x L_y}, \tag{8}$$

where the factor of 2 in the denominator accounts for the periodic setup. Considering a reference layer, this is intuitively clear, because half the supplied heat will flow to the other reservoir in the reference box and the other half to its image in the neighbouring box. The heat flux is an input parameter of the HEX algorithm, which we set to 0.15 in reduced units.

The dependence of the energy loss at the final time on the timestep is shown in Fig. 2. From the quadratic fit, it is clear that the HEX algorithm exhibits an energy drift which scales as $\mathcal{O}(\Delta t^2)$. On the other hand, the energy was conserved perfectly well in *NVE* simulations at the peak temperature inside the hot reservoir. (The temperature profiles are discussed in Sec. VI.)

III. EQUATIONS OF MOTION

To gain a better understanding of the energy drift of the HEX algorithm, we first derive the ordinary differential equations (ODEs) solved by the algorithm in the limit $\Delta t \to 0$. To this end we consider the velocity update for continuous time. Dropping all particle and region indices for readability and eliminating the intermediate velocities, we can cast Eq. (4f) into

$$\begin{aligned}\boldsymbol{v}^{n+1} = \;& \bar{\xi}^{n+1}\left[\xi^{n}\boldsymbol{v}^{n} + \left(1-\xi^{n}\right)\boldsymbol{v}_{\Gamma}^{n} + \frac{\Delta t}{2m}\left(\boldsymbol{f}^{n}+\boldsymbol{f}^{n+1}\right)\right] \\ & + \left(1-\bar{\xi}^{n+1}\right)\bar{\boldsymbol{v}}_{\Gamma}^{n+1}. \end{aligned} \tag{9}$$

If we subtract $\boldsymbol{v}^n$ on both sides and divide by the timestep, we get

$$\begin{aligned}\frac{\boldsymbol{v}^{n+1}-\boldsymbol{v}^{n}}{\Delta t} = \;& \frac{\bar{\xi}^{n+1}}{2m}\left(\boldsymbol{f}^{n}+\boldsymbol{f}^{n+1}\right) \\ + \;& \frac{\left(\bar{\xi}^{n+1}\xi^{n}-1\right)\boldsymbol{v}^{n}}{\Delta t} \\ + \;& \frac{\bar{\xi}^{n+1}\left(1-\xi^{n}\right)\boldsymbol{v}_{\Gamma}^{n} + \left(1-\bar{\xi}^{n+1}\right)\bar{\boldsymbol{v}}_{\Gamma}^{n+1}}{\Delta t}. \end{aligned} \tag{10}$$

It is straightforward to show that

$$\frac{\left(\bar{\xi}^{n+1}\xi^{n}-1\right)\boldsymbol{v}^{n}}{\Delta t} \to \frac{\mathcal{F}_{\Gamma}\boldsymbol{v}(t^{n})}{2\mathcal{K}_{\Gamma}(t^{n})}$$

and

$$\frac{\bar{\xi}^{n+1}\left(1-\xi^{n}\right)\boldsymbol{v}_{\Gamma}^{n} + \left(1-\bar{\xi}^{n+1}\right)\bar{\boldsymbol{v}}_{\Gamma}^{n+1}}{\Delta t} \to -\frac{\mathcal{F}_{\Gamma}\boldsymbol{v}_{\Gamma}(t^{n})}{2\mathcal{K}_{\Gamma}(t^{n})}$$

in the limit of $\Delta t \to 0$. From Eq. (4c), it is immediately obvious that the derivative of the coordinates is given by the velocities.

The continuous equations of motion solved by the HEX algorithm are therefore given by

$$\dot{\boldsymbol{r}}_i = \boldsymbol{v}_i, \tag{11a}$$

$$\dot{\boldsymbol{v}}_i = \frac{\boldsymbol{f}_i}{m_i} + \frac{\boldsymbol{\eta}_i}{m_i}, \tag{11b}$$

where the thermostatting force is defined as

$$\boldsymbol{\eta}_i = \begin{cases} m_i \frac{\mathcal{F}_{\Gamma_{k(\boldsymbol{r}_i)}}}{2\mathcal{K}_{\Gamma_{k(\boldsymbol{r}_i)}}}\left(\boldsymbol{v}_i - \boldsymbol{v}_{\Gamma_{k(\boldsymbol{r}_i)}}\right) & \text{if } k(\boldsymbol{r}_i) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

In order for the equations to be well-defined, we assume that there are sufficiently many particles inside any reservoir, i.e. regions with $k(\boldsymbol{r}_i) > 0$, such that the non-translational kinetic energy never vanishes. Outside the reservoirs the thermostatting force is zero and the particles obey Hamiltonian motion. Some further properties of the equations are analysed in Appendix A.

IV. OPERATOR SPLITTING

Our goal is to show that the energy drift is caused by higher-order truncation terms, which are not taken into account in the time integration. These terms can be derived easily once the integration scheme is expressed in terms of a Trotter factorisation of the Liouville operator.

A. Trotter factorisation

Tuckerman *et al.*[19] showed that reversible integrators can be generated based on a Trotter factorisation of the Liouville operator iL. Utilising the same theoretical framework, we consider the splitting

$$iL = iL_1 + iL_2, \tag{13a}$$

$$iL_1 = \sum_{j=1}^{N}\sum_{\alpha\in\{x,y,z\}} \frac{\eta_{j,\alpha}}{m_j}\frac{\partial}{\partial v_{j,\alpha}}, \tag{13b}$$

$$iL_2 = \sum_{j=1}^{N}\sum_{\alpha\in\{x,y,z\}} \left[\frac{f_{j,\alpha}}{m_j}\frac{\partial}{\partial v_{j,\alpha}} + v_{j,\alpha}\frac{\partial}{\partial r_{j,\alpha}}\right] \tag{13c}$$

and apply it to the current state of the system which is fully described by $\boldsymbol{x}$ in the $6N$-dimensional phase space. The exact time evolution of the system is formally given by

$$\boldsymbol{x}_{\text{ex}}(t) = \mathrm{e}^{tiL}\boldsymbol{x}(0). \tag{14}$$

Unfortunately, it is not feasible to evaluate this expression analytically for the equations we are interested in. The problem can be simplified, however, by considering the approximation

$$\boldsymbol{x}(t) = \left[\mathrm{e}^{\frac{\Delta t}{2}iL_1}\mathrm{e}^{\Delta t iL_2}\mathrm{e}^{\frac{\Delta t}{2}iL_1}\right]^{P}\boldsymbol{x}(0), \tag{15}$$

where P is an integral number which implicitly defines the timestep through $\Delta t = t/P$. The operator $\mathrm{e}^{\frac{\Delta t}{2}iL_1}$ acts on the velocities and adds the energy $\Delta Q/2$ to the system. In fact, as shown in Appendix A, the velocity update of the HEX algorithm is the exact solution of this operation. Hamilton's equations of motion are then integrated with $\mathrm{e}^{\Delta t iL_2}$ followed by the second energy supply. For the analysis in the next subsection, we assume that all operations in Eq. (15) can be carried out analytically, although in the simulation we use an additional approximation of $\mathrm{e}^{\Delta t iL_2}$, as discussed in Sec. V.

B. Local truncation error

The splitting given by Eq. (15) is known as Strang splitting[20]. It has the local truncation error[21]

$$\boldsymbol{x}(\Delta t) - \boldsymbol{x}_{\text{ex}}(\Delta t) = \Delta t^{3}\mathcal{E}\boldsymbol{x}_{\text{ex}}(0) + \mathcal{O}\left(\Delta t^{4}\right), \tag{16}$$

where the first term on the RHS is determined by the operator

$$\mathcal{E} = \frac{1}{12}\left[iL_2,\left[iL_2,iL_1\right]\right] - \frac{1}{24}\left[iL_1,\left[iL_1,iL_2\right]\right] \tag{17}$$

and $[A, B] = AB - BA$ is the commutator. Rearranging terms, we find

$$\boldsymbol{x}(\Delta t) - \Delta t^3 \mathcal{E} \boldsymbol{x}_{\text{ex}}(0) = \boldsymbol{x}_{\text{ex}}(\Delta t) + \mathcal{O}\left(\Delta t^4\right). \tag{18}$$

This means that the key to improving the accuracy of the numerical approximation is to apply the correction $-\Delta t^3 \mathcal{E} \boldsymbol{x}_{\text{ex}}(0)$ to the original solution. Alternatively, we can also use a correction $-\Delta t^3 \mathcal{E} \tilde{\boldsymbol{x}}(\Delta t)$, where $\tilde{\boldsymbol{x}}(\Delta t) = \boldsymbol{x}_{\text{ex}}(0) + \mathcal{O}(\Delta t)$, without changing the order of the truncation error.

V. ENHANCED HEX ALGORITHM

The analysis of the previous section remains valid for any approximation of $\mathrm{e}^{\Delta t i L_2}$ which is sufficiently accurate. This is necessarily the case if the local truncation error is $\mathcal{O}\left(\Delta t^4\right)$ or higher. Velocity Verlet integration is less accurate than that and has a local truncation error of $\mathcal{O}\left(\Delta t^3\right)$. Nevertheless, we found that it is fully sufficient to consider a coordinate correction of the form of Eq. (18) to get hold of the energy loss. We therefore ignored the additional velocity Verlet truncation error and all other correction terms in Eq. (18) affecting velocities only.

This analysis leads us directly to the *enhanced heat exchange (eHEX)* algorithm, which is defined through the update sequence

$$\begin{aligned}
\bar{\boldsymbol{v}}_i^n &= \xi_{k(\boldsymbol{r}_i)}^n \boldsymbol{v}_i^n + \left(1 - \xi_{k(\boldsymbol{r}_i)}^n\right) \boldsymbol{v}_{\Gamma_{k(\boldsymbol{r}_i)}}^n, && \text{(19a)}\\
\bar{\boldsymbol{v}}_i^{n+\frac{1}{2}} &= \bar{\boldsymbol{v}}_i^n + \frac{\Delta t}{2m_i} \boldsymbol{f}_i^n, && \text{(19b)}\\
\bar{\boldsymbol{r}}_i^{n+1} &= \boldsymbol{r}_i^n + \Delta t\, \bar{\boldsymbol{v}}_i^{n+\frac{1}{2}}, && \text{(19c)}\\
\boldsymbol{f}_i^{n+1} &= -\nabla_{\boldsymbol{r}_i} U(\boldsymbol{r}) \,|_{\boldsymbol{r}=\bar{\boldsymbol{r}}^{n+1}}, && \text{(19d)}\\
\bar{\boldsymbol{v}}_i^{n+1} &= \bar{\boldsymbol{v}}_i^{n+\frac{1}{2}} + \frac{\Delta t}{2m_i} \boldsymbol{f}_i^{n+1}, && \text{(19e)}\\
\boldsymbol{v}_i^{n+1} &= \bar{\xi}_{k(\bar{\boldsymbol{r}}_i)}^{n+1} \bar{\boldsymbol{v}}_i^{n+1} + \left(1 - \bar{\xi}_{k(\bar{\boldsymbol{r}}_i)}^{n+1}\right) \bar{\boldsymbol{v}}_{\Gamma_{k(\bar{\boldsymbol{r}}_i)}}^{n+1}, && \text{(19f)}\\
\boldsymbol{r}_i^{n+1} &= \bar{\boldsymbol{r}}_i^{n+1} - \Delta t^3 \mathcal{E} \bar{\boldsymbol{r}}_i^{n+1}. && \text{(19g)}
\end{aligned}$$

Apart from the last integration step and some relabelling, this scheme is identical to the HEX algorithm. As shown in Appendix A, the correction term is given by

$$\begin{aligned}
\mathcal{E} r_{i,\alpha} = &\frac{\eta_{i,\alpha}}{m_i \mathcal{K}_{\Gamma_{k(\boldsymbol{r}_i)}}} \left[\frac{\mathcal{F}_{\Gamma_{k(\boldsymbol{r}_i)}}}{48} + \frac{1}{6} \sum_{j \in \gamma_{k(\boldsymbol{r}_i)}} \boldsymbol{f}_j \cdot \left(\boldsymbol{v}_j - \boldsymbol{v}_{\Gamma_{k(\boldsymbol{r}_i)}} \right) \right] \\
&- \frac{\mathcal{F}_{\Gamma_{k(\boldsymbol{r}_i)}}}{12 \mathcal{K}_{\Gamma_{k(\boldsymbol{r}_i)}}} \left[\frac{f_{i,\alpha}}{m_i} - \frac{1}{m_{\Gamma_{k(\boldsymbol{r}_i)}}} \sum_{j \in \gamma_{k(\boldsymbol{r}_i)}} f_{j,\alpha} \right]
\end{aligned} \tag{20}$$

and evaluated at the state $\bar{\boldsymbol{x}}^{n+1}$. We note that this expression vanishes for particles outside any reservoir, because the thermostatting force is zero in that case. The scaling factors $\xi_{k(\boldsymbol{r}_i)}^n$ and $\bar{\xi}_{k(\bar{\boldsymbol{r}}_i)}^{n+1}$ are calculated at the system states $\boldsymbol{x}^n$ and $\bar{\boldsymbol{x}}^{n+1}$, respectively. As in the formulation of the original algorithm, we also consider the case where all the energy is supplied asymmetrically in Eq. (19f). We refer to this version of the algorithm as eHEX/a.

A. Rigid molecules

Employing constraining forces, we can extend the eHEX algorithm to a system of rigid bodies, such as SPC/E water[22]. In the SHAKE algorithm[23], originally devised for Verlet integration, rigidity is imposed by solving iteratively for a set of Lagrange multipliers. If the underlying equations are integrated with the velocity Verlet algorithm, a second set of constraining forces is required to eliminate velocity components along any fixed bond. This is taken into account by the RATTLE algorithm[24] which we implemented in LAMMPS.

To be compatible with the treatment of constraining forces in LAMMPS, we consider the eHEX/a algorithm for rigid bodies. We use RATTLE to ensure that the velocities and positions are satisfied up to the target tolerance after the second velocity update (Eq. (19e)). Provided that all sites of a reference molecule are located in the same region, the scaling and shifting in Eq. (19f) does not violate the constraints. For this reason, we only rescale an individual site of a molecule if its centre of mass is located within the reservoir.

For the small fraction of molecules inside a reservoir, the coordinate correction in Eq. (19g) introduces an $\mathcal{O}(\Delta t^3)$ error in the bond distances. This error is small and of the same order as the local error of the RATTLE algorithm itself[24]. For this reason, we consider an unconstrained update acceptable. However, we monitored the maximum relative errors throughout all simulations. The constraining forces for the coordinates are recalculated at the end of the timestep to ensure that the positions are correct after the subsequent velocity Verlet update.

B. Model system

In addition to the monatomic system, we tested the eHEX/a algorithm for the SPC/E water model. The simulation box with dimensions $L_z/2 = L_x = L_y = 25.26$ Å contained 1024 molecules resulting in a density of 0.95 g/cm^3. We used a real-space cutoff of 11 Å for the LJ and Coulomb interactions, which were evaluated with standard Ewald summation[15]. The damping parameter was $\alpha = 6.816/L_x$ with 9841 $\boldsymbol{k}$-vectors (before employing symmetry properties of the reciprocal sum).

Starting from a lattice structure, we rescaled velocities for 10 ps to drive the system close to a target temperature of 400 K. We employed a timestep of 1 fs and the equations were integrated with velocity Verlet. This was followed by a 2×10^5 timestep *NVT* simulation using a Nosé–Hoover thermostat with a relaxation time of 1 ps. The total energy of the last configuration was then adjusted such that it corresponds to the average of the *NVT* run. The average temperature over a subsequent 2×10^5 timestep *NVE* simulation was (400.5 ± 0.2) K.

We then switched on the thermostat and waited for 100 ps for the system to reach a steady state before starting with the 1 ns production run. The reservoirs of width 4 Å were centred at the points $z = \pm L_z/4$ and we imposed a heat flux of 4.08×10^{10} W/m^2. As a reference for the energy conservation, we carried out an additional set of 1 ns *NVE* simulations at various timesteps. The temperature in these runs was (468.0 ± 0.2) K.

VI. RESULTS

The effect of the additional coordinate integration in the eHEX algorithm on the total energy conservation is shown in Figs. 3–4. As can be seen, the new algorithm exhibits excellent energy conservation. Even for large timesteps close to the stability limit for a *NVE* simulation at the peak temperature ($\Delta t^*_{\text{max, LJ}} \approx 0.0075$ and $\Delta t_{\text{max, SPC/E}} \approx 3.5$ fs), there is no noticeable drift on this scale. The energy loss of the HEX algorithm, on the other hand, is substantial. At the largest timestep, the total system energy changed by about 0.45% for LJ and 1.6% for SPC/E, respectively. Although an energy loss of several percent was considered acceptable in the past[6,10,11], it sets an upper limit to the accessible simulation time scales. The only way to circumvent this problem apart from coupling the system to an additional thermostat is to decrease the timestep and thereby waste valuable computing time. Based on a series of eight simulations at the largest timestep and with different initial conditions, we can give a conservative estimate of the improvement due to the new algorithms. For LJ we found that the eHEX algorithm loses at least 500 times less energy than the HEX algorithm (450 for eHEX/a as compared to HEX/a). For SPC/E water we found that the eHEX/a algorithm improves the energy conservation by at least a factor of 100 as compared to the HEX/a algorithm. The accessible simulation time scale therefore increased by two orders of magnitude.

The spatial variation in temperature is shown in Figs. 5–6. For the monatomic system, the results agree well without any marked differences (Fig. 5). We note that in both cases there are some visible discontinuities in the vicinity of the reservoirs. This should not be very surprising, since the thermostatting force is also discontinuous. We found that the gap decreases as we go to lower temperature gradients, because the fluid can dissipate the heat sufficiently fast. A possible way of controlling this gap is to employ a weight function and to redefine temperature such that it is continuous at the boundary of the reservoir[4]. This procedure allows for better control and is numerically convenient, but it is not obvious which weight function is physically most meaningful. Furthermore, generalising this approach to arbitrary reservoir shapes is challenging, because it requires some sort of signed distance information to the boundary.

For SPC/E water the energy loss at large timesteps is reflected in a slight drop of temperature (Fig. 6). The overall profiles agree well, but they are shifted by a few Kelvin. This shift is consistent with the energy loss of about 1% for the 2.5 fs timestep. There are no visible temperature discontinuities in the vicinity of the reservoirs. This might be related to the fact that in our scheme the boundaries are naturally smeared out as we only rescale entire molecules which could be intersected by the reservoir boundary.

Although we omitted a constrained coordinate update in the eHEX/a algorithm, the relative deviation from the ideal bond distance never exceeded 1.1×10^{-5}. This was the case for the largest timestep of 3 fs, but the error decays rapidly (with Δt^3) such that it reduced to 3.6×10^{-7} for a timestep of 1 fs. The maximum induced relative velocity along any rigid bond was an order of magnitude lower for both timesteps, respectively. Only a small fraction of molecules inside a reservoir ($\approx 16\%$) suffers from this inconsistency. We consider this error acceptable and an unconstrained update justified. An extension of the eHEX algorithm to a constrained update is possible in case higher precision is required.

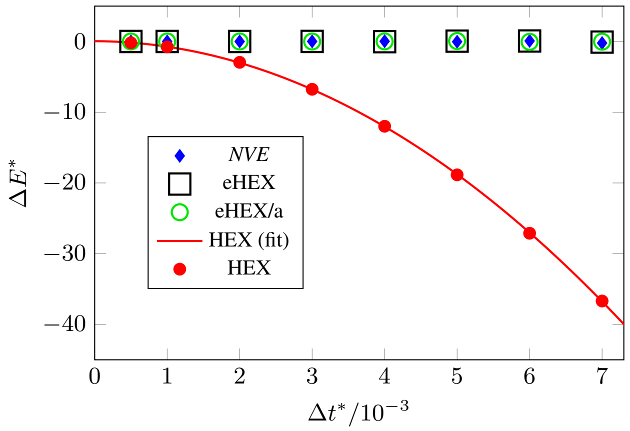

FIG. 3. Energy loss for LJ at the final time $t^* = 5000$ for various timesteps. The equilibrium run (blue diamonds) and the symmetric (black squares) and asymmetric (green, open circles) versions of the eHEX algorithm, respectively, do not show any appreciable drift. The energy loss of the HEX algorithm (red, full circles) together with a quadratic fit (red, solid line) is shown for comparison.

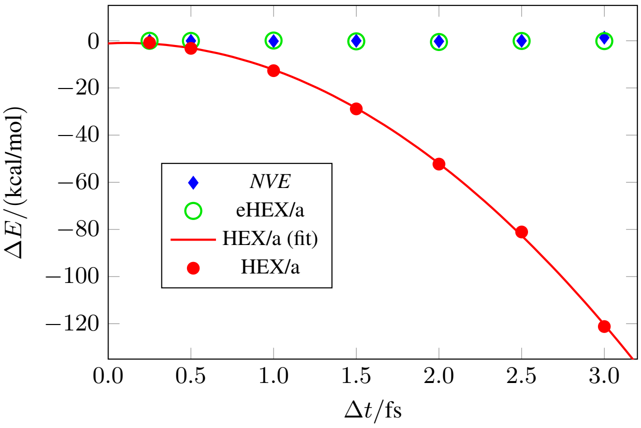

FIG. 4. Energy loss for SPC/E water at the final time $t = 1$ ns for various timesteps. The equilibrium run (blue diamonds) is compared to the asymmetric eHEX algorithm (green, open circles) and the asymmetric HEX algorithm (red, full circles) together with a quadratic fit (red, solid line).

With regard to conservation of total linear momentum, we found that both algorithms satisfied this condition perfectly. We initialised the linear momentum of the box to zero at the beginning and it remained close to machine precision throughout the entire simulation.

VII. CONCLUSIONS

In this paper, we have presented a new algorithm for NEMD simulations of thermal gradients. The method comprises an extension to the HEX algorithm, which rescales and shifts velocities of particles inside reservoirs to impose a constant

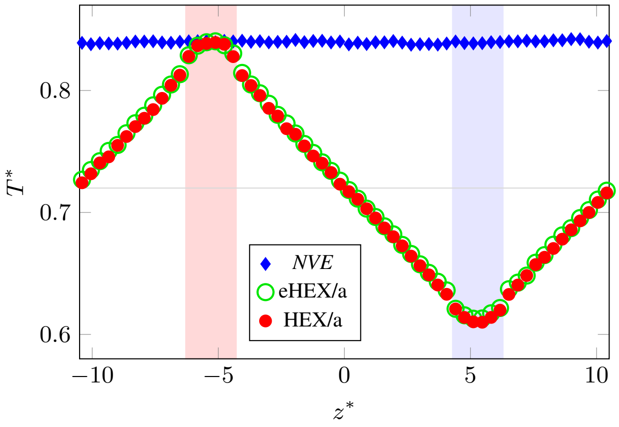

FIG. 5. Comparison of the temperature profiles for LJ.

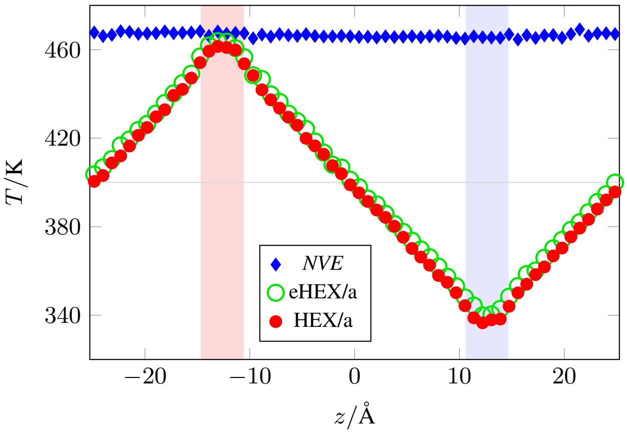

FIG. 6. Comparison of the temperature profiles for SPC/E water.

heat flux. The problem with the original algorithm is that it exhibits a drift in the total energy whose origin remained hitherto unclear. For long simulations, this energy loss becomes restrictive, limiting the accessible simulation time scales to a few nanoseconds. In our approach, we reformulated the HEX algorithm as a Trotter factorisation of the Liouville operator. Using this theoretical framework, it is straightforward to determine higher-order truncation terms which are a consequence of the employed operator splitting. We demonstrated that the leading-order truncation error of the coordinates is responsible for the observed energy drift.

To test the accuracy of the method, we implemented the eHEX algorithm in LAMMPS and ran simulations on a Lennard-Jones system and SPC/E water. In both cases, we observed at least a hundredfold reduction in the energy loss as compared to the HEX algorithm. With the eHEX algorithm, it is therefore possible to carry out constant heat flux simulations which are on the order of a hundred nanoseconds and based on fully deterministic equations of motion.

ACKNOWLEDGMENTS

PW gratefully acknowledges stimulating discussions with Chongli Qin, Clemens Moritz, Raman Ganti and Aleks Reinhardt, as well as financial support through a DOC Fellowship of the Austrian Academy of Sciences. PW would also like to thank members of the LAMMPS mailing list, especially Paul Crozier, Axel Kohlmeyer and Steve Plimpton, for helpful discussions. Further financial support from the Federation of Austrian Industry (IV) Carinthia and the Austrian Science Fund FWF within the SFB Vicom (project F41) is acknowledged with gratitude. The results presented here have been achieved in part using the Vienna Scientific Cluster (VSC). DF acknowledges support from Engineering and Physical Sciences Research Council Programme Grant EP/I001352/1. Additional data related to this publication are available at the University of Cambridge data repository (https://www.repository.cam.ac.uk/handle/1810/250539).

Appendix A: Heat exchange algorithm

1. Exact solution

We would like to show that the rescaling step

$$\boldsymbol{v}_i(t) = \xi \boldsymbol{v}_i(0) + \big(1-\xi\big)\boldsymbol{v}_\Gamma(0) \tag{A1}$$

of the HEX algorithm is the exact solution of

$$\dot{\boldsymbol{v}}_i = \frac{\mathcal{F}}{2\mathcal{K}}(\boldsymbol{v}_i - \boldsymbol{v}_\Gamma), \tag{A2}$$

where $\mathcal{K}$ is given by Eq. (1) and ξ by Eq. (3). We will first show that $\mathcal{F}/2\mathcal{K}$ is independent of the particle velocities and only a function of time. This can be seen easily by considering the time evolution of the internal kinetic energy, which is given by

$$\begin{aligned}\frac{\mathrm{d}\mathcal{K}}{\mathrm{d}t} &= \sum_{i\in\gamma} m_i\left(\boldsymbol{v}_i - \boldsymbol{v}_\Gamma\right)\cdot\left(\dot{\boldsymbol{v}}_i - \dot{\boldsymbol{v}}_\Gamma\right) && \text{(A3a)}\\ &= \mathcal{F} && \text{(A3b)}\end{aligned}$$

and therefore we can write $\mathcal{K}(t) = \mathcal{K}(0) + \mathcal{F}t$. We note that we can exchange the order of taking the time derivative and the summation, because the particle positions are fixed during this operation. At the same time, it is easy to see that the centre of mass velocity is constant in time since

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{v}_\Gamma}{\mathrm{d}t} &= \frac{1}{m_\Gamma}\sum_{i\in\gamma} m_i \dot{\boldsymbol{v}}_i && \text{(A4)}\\ &= \frac{1}{m_\Gamma}\sum_{i\in\gamma} m_i\left[\frac{\mathcal{F}}{2\mathcal{K}}\left(\boldsymbol{v}_i - \boldsymbol{v}_\Gamma\right)\right] && \text{(A5)}\\ &= 0. && \text{(A6)}\end{aligned}$$

In order to solve Eq. (A2) analytically, it is advantageous to carry out a variable transformation first. Let us consider the transformation $\boldsymbol{v}_i \mapsto \bar{\boldsymbol{v}}_i = \boldsymbol{v}_i - \boldsymbol{v}_\Gamma$. With the definition

$$\lambda(t) = \frac{\mathcal{F}}{2\left[\mathcal{K}(0) + \mathcal{F}t\right]}, \tag{A7}$$

we can express the time evolution of the new velocities as

$$\dot{\bar{\boldsymbol{v}}}_i = \lambda \bar{\boldsymbol{v}}_i. \tag{A8}$$

The solution of this equation is then given by

$$\bar{\boldsymbol{v}}_i(t) = \mathrm{e}^{\int_0^t \mathrm{d}t' \, \lambda(t')} \bar{\boldsymbol{v}}_i(0) = \xi \bar{\boldsymbol{v}}_i(0). \tag{A9}$$

If we substitute the old variables back, we recover Eq. (A1) which proves the assertion.

2. Splitting error

In this section, we sketch the derivation of the leading-order error term of the coordinate integration arising from the operator splitting in the HEX algorithm. To this end, we evaluate the expression

$$\mathcal{E}r_{i,\alpha} = \tag{A10}$$
$$\left(\frac{1}{12} \big[iL_2, [iL_2, iL_1] \big] - \frac{1}{24} \big[iL_1, [iL_1, iL_2] \big] \right) r_{i,\alpha}$$

for the operators

$$iL_1 = \sum_{j=1}^{N} \sum_{\beta \in \{x,y,z\}} \frac{\eta_{j,\beta}}{m_j} \frac{\partial}{\partial v_{j,\beta}}, \tag{A11a}$$

$$iL_2 = \sum_{j=1}^{N} \sum_{\beta \in \{x,y,z\}} \left[\frac{f_{j,\beta}}{m_j} \frac{\partial}{\partial v_{j,\beta}} + v_{j,\beta} \frac{\partial}{\partial r_{j,\beta}} \right]. \tag{A11b}$$

For the first term in Eq. (A10) we find

$$\big[iL_2, [iL_2, iL_1] \big] r_{i,\alpha}$$
$$= -2 \sum_{j=1}^{N} \sum_{\beta} \left[\frac{f_{j,\beta}}{m_j} \frac{\partial}{\partial v_{j,\beta}} + v_{j,\beta} \frac{\partial}{\partial r_{j,\beta}} \right] \frac{\eta_{i,\alpha}}{m_i} \tag{A12a}$$
$$= -\frac{2}{m_i} \sum_{j \in \gamma_{k(\boldsymbol{r}_i)}} \sum_{\beta} \frac{f_{j,\beta}}{m_j} \frac{\partial \eta_{i,\alpha}}{\partial v_{j,\beta}}, \tag{A12b}$$

omitting summation bounds for β for readibility. In the last step we assumed that particles do not cross reservoir boundaries, in which case $\eta_{i,\alpha}$ depends only on the velocities of particles within the reservoir $\Gamma_{k(\boldsymbol{r}_i)}$. For the second term in Eq. (A10) we find

$$\big[iL_1, [iL_1, iL_2] \big] r_{i,\alpha} = \frac{1}{m_i} \sum_{j \in \gamma_{k(\boldsymbol{r}_i)}} \sum_{\beta} \frac{\eta_{j,\beta}}{m_j} \frac{\partial \eta_{i,\alpha}}{\partial v_{j,\beta}} \tag{A13}$$

and combining the two expressions we get

$$\mathcal{E}r_{i,\alpha} = -\frac{1}{6m_i} \sum_{j \in \gamma_{k(\boldsymbol{r}_i)}} \sum_{\beta} \frac{1}{m_j} \left(f_{j,\beta} + \frac{\eta_{j,\beta}}{4} \right) \frac{\partial \eta_{i,\alpha}}{\partial v_{j,\beta}}. \tag{A14}$$

It is straightforward to compute the derivative

$$\frac{\partial \eta_{i,\alpha}}{\partial v_{j,\beta}} = \frac{m_i \mathcal{F}_{\Gamma_{k(\boldsymbol{r}_i)}}}{2\mathcal{K}_{\Gamma_{k(\boldsymbol{r}_i)}}} \left[\delta_{\alpha,\beta} \left(\delta_{i,j} - \frac{m_j}{m_{\Gamma_{k(\boldsymbol{r}_i)}}} \right) \right. \tag{A15}$$
$$\left. - \frac{m_j}{\mathcal{K}_{\Gamma_{k(\boldsymbol{r}_i)}}} \left(v_{j,\beta} - v_{\Gamma_{k(\boldsymbol{r}_i)},\beta} \right) \left(v_{i,\alpha} - v_{\Gamma_{k(\boldsymbol{r}_i)},\alpha} \right) \right],$$

where $\delta_{i,j}$ is the Kronecker delta. The final result, Eq. (20), is then recovered by substituting the derivative in Eq. (A14) with the expression above.

[1] D. J. Evans and W. G. Hoover, Annu. Rev. Fluid Mech. **18**, 243 (1986).

[2] D. J. Evans, Phys. Lett. A **91**, 457 (1982).

[3] W. T. Ashurst, in *Advances in Thermal Conductivity: 13th International Conference on Thermal Conductivity, Lake Ozark, Nov. 1973, Papers* (University of Missouri-Rolla, Rolla, MO, 1976) pp. 89–98.

[4] A. Baranyai, Phys. Rev. E **54**, 6911 (1996).

[5] G. Ciccotti and A. Tenenbaum, J. Stat. Phys. **23**, 767 (1980).

[6] T. Ikeshoji and B. Hafskjold, Mol. Phys. **81**, 251 (1994).

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

[8] S. Kuang and J. D. Gezelter, J. Chem. Phys. **133**, 164101 (2010).

[9] S. Kuang and J. D. Gezelter, Mol. Phys. **110**, 691 (2012).

[10] B. Hafskjold, T. Ikeshoji, and S. K. Ratkje, Mol. Phys. **80**, 1389 (1993).

[11] F. Bresme, B. Hafskjold, and I. Wold, J. Phys. Chem. **100**, 1879 (1996).

[12] M. Bugel and G. Galliero, Chem. Phys. **352**, 249 (2008).

[13] S. Aubry, D. J. Bammann, J. J. Hoyt, R. E. Jones, C. J. Kimmer, P. A. Klein, G. J. Wagner, E. B. Webb III, and J. A. Zimmerman, "A robust, coupled approach for atomistic-continuum simulation," Tech. Rep. (Sandia National Laboratories, 2004).

[14] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).

[15] D. Frenkel and B. Smit, *Understanding Molecular Simulation*, 2nd ed. (Academic Press, San Diego, 2002).

[16] S. Toxvaerd and J. C. Dyre, J. Chem. Phys. **134**, 081102 (2011).

[17] S. Nosé, J. Chem. Phys. **81**, 511 (1984).

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

[19] M. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **97**, 1990 (1992).

[20] G. Strang, SIAM J. Numer. Anal. **5**, 506 (1968).

[21] R. J. LeVeque and J. Oliger, Math. Comput. **40**, 469 (1983).

[22] H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma, J. Phys. Chem. **91**, 6269 (1987).

[23] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[24] H. C. Andersen, J. Comput. Phys. **52**, 24 (1983).

[25] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, J. Chem. Phys. **76**, 637 (1982).