

Assessing numerical methods for molecular and particle simulation

Shang, Xiaocheng; Kröger, Martin; Leimkuhler, Benedict

DOI:

[10.1039/C7SM01526G](https://doi.org/10.1039/C7SM01526G)

License:

Creative Commons: Attribution (CC BY)

Document Version

Publisher's PDF, also known as Version of record

Citation for published version (Harvard):

Shang, X, Kröger, M & Leimkuhler, B 2017, 'Assessing numerical methods for molecular and particle simulation', *Soft Matter*, vol. 13, no. 45, pp. 8565-8578. <https://doi.org/10.1039/C7SM01526G>

[Link to publication on Research at Birmingham portal](#)

Publisher Rights Statement:

Checked for eligibility: 25/07/2019

Shang, Xiaocheng, Martin Kröger, and Benedict Leimkuhler. "Assessing numerical methods for molecular and particle simulation." *Soft matter* 13.45 (2017): 8565-8578. [10.1039/C7SM01526G](https://doi.org/10.1039/C7SM01526G)

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.



Cite this: *Soft Matter*, 2017, 13, 8565

Assessing numerical methods for molecular and particle simulation

Xiaocheng Shang,  *^a Martin Kröger  ^a and Benedict Leimkuhler  *^b

We discuss the design of state-of-the-art numerical methods for molecular dynamics, focusing on the demands of soft matter simulation, where the purposes include sampling and dynamics calculations both in and out of equilibrium. We discuss the characteristics of different algorithms, including their essential conservation properties, the convergence of averages, and the accuracy of numerical discretizations. Formulations of the equations of motion which are suited to both equilibrium and nonequilibrium simulation include Langevin dynamics, dissipative particle dynamics (DPD), and the more recently proposed “pairwise adaptive Langevin” (PAaL) method, which, like DPD but unlike Langevin dynamics, conserves momentum and better matches the relaxation rate of orientational degrees of freedom. PAaL is easy to code and suitable for a variety of problems in nonequilibrium soft matter modeling; our simulations of polymer melts indicate that this method can also provide dramatic improvements in computational efficiency. Moreover we show that PAaL gives excellent control of the relaxation rate to equilibrium. In the nonequilibrium setting, we further demonstrate that while PAaL allows the recovery of accurate shear viscosities at higher shear rates than are possible using the DPD method at identical timestep, it also outperforms Langevin dynamics in terms of stability and accuracy at higher shear rates.

Received 31st July 2017,
Accepted 20th October 2017

DOI: 10.1039/c7sm01526g

rsc.li/soft-matter-journal

1 Introduction

In this article, we provide a current and detailed perspective on the design of stochastic methods for simulating molecular and particle systems. Most of our discussion is general and equally applicable to simple and complex molecular fluids and polymer solutions, and to both equilibrium and nonequilibrium modeling. Modern software packages such as LAMMPS¹ offer a bewildering array of options for particle simulation, including choices regarding the model ensemble, equations of motion, discretization method, and parameter selection. In this article we contrast a number of the different schemes available, drawing on recent advances in the literature and focussing on the practical needs of the simulation community. All of the existing methods are convergent in the sense that, for suitable choice of parameters and in the limit of small timestep, they are capable of reproducing the exact statistical properties with high accuracy; however, the methods have very different computational efficiencies. In practice, the choice of method can mean the difference between a computational task completing in a day or a week. The challenge of designing efficient methods

is particularly acute in nonequilibrium modeling, where the lack of a simple known form for the invariant distribution makes benchmarking challenging and where the delicate approximation of dynamical behavior plays an important role.

The state-of-the-art in molecular dynamics and its limitations are well documented in recent reviews.^{2,3} Let us briefly review the challenges of simulation of polymeric systems,^{4–7} which constitute a broad area of research of pharmaceutical, materials, chemical, biological and physical relevance, and an area where simulation times easily exceed available resources. While the system size required to avoid significant finite-size effects scales, with the polymerization degree N , as $N^{3/2}$ for flexible chains and as N^3 for stiff chains, respectively, the longest relaxation time scales as N^2 for dilute systems and as N^3 (or larger) for concentrated and entangled systems, respectively; the total simulation time is thus $\sim N^6$ for a concentrated polymer solution, while a typical polymerization degree is $N \approx 10^4$ – 10^5 for a synthetic polymer or a biopolymer like hyaluronan. Such systems can be investigated qualitatively using coarse-graining strategies and multiscale modeling approaches of various kinds.^{8,9} However, atomistic simulation of polymeric systems is limited to the study of a few molecules, or concentrated, eventually semicrystalline systems containing simple polymers like polyethylene with $N \leq 2000$ over a duration of a few tens of nanoseconds^{10–14} and subject to deformation or flow.^{15–17} Highly entangled polymeric systems

^a Department of Materials, Polymer Physics, ETH Zürich, CH-8093 Zürich, Switzerland. E-mail: x.shang@mat.ethz.ch

^b School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh, Edinburgh, EH9 3FD, UK. E-mail: b.leimkuhler@ed.ac.uk



with practically relevant N , and their confined counterparts like polymer brushes^{5,18–20} are still out of reach for atomistic simulations. The limitations are even more severe for polymers in nanocomposites,^{6,21,22} branched or hyperbranched polymers like dendronized polymers,^{20,23–26} and polyelectrolytes.^{27–29}

This article is addressed to stochastic simulation techniques for polymer models based on generalizations of Brownian or Langevin dynamics. The challenges of simulation are well exemplified by two model polymer systems: (i) the single polymer chain in implicit solvent, and (ii) a polymer melt. Both models combine aspects of sampling and dynamical approximation. In the case of a single polymer, the dynamics of the thermostat plays a crucial role in describing the relaxation behavior. It is important in this setting to mimic the underlying internal dynamical processes of the molecule while correctly modeling the exchange of energy between polymer and bath. For the melt, the key difficult quantities are rheological properties like shear viscosity or normal stress difference and disentanglement time, and the system dynamics is typically dominated by the frequent collisions of particles. We restrict our attention to the polymer melt case in this article, because it is computationally more demanding.

Designing effective algorithms involves the selection of a formulation or modeling framework, choice of parameterization, and design of numerical discretization. These choices cannot be isolated from each other. In practice the form of the numerical discretization is strongly dependent on the formulation used, and the choice of parameterization will depend on both the goals of simulation and the numerical scheme.

The common feature of all the most popular formulations in use is that they are designed to facilitate sampling in the canonical ensemble. That is, when applied to a conservative system with a total number of N_t particles and Hamiltonian energy function $H = \sum_{i=1}^{N_t} \mathbf{p}_i \cdot \mathbf{p}_i / (2m) + U(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{N_t})$, they are designed to drive the system toward the canonical equilibrium state with probability density

$$\rho_\beta(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{N_t}, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{N_t}) = Z^{-1} \exp(-\beta H)$$

where Z is a suitable normalizing constant (*i.e.*, the partition function), $\beta = (k_B T)^{-1}$, with k_B being the Boltzmann constant, T the system temperature, and m the assumed uniform mass of the N_t particles. (Generalizing all results of this article to nonuniform particle masses would be straightforward but the uniform mass simplifies presentation of some formulas.) Temperature control is crucial for NVT simulation, but also plays a fundamental role in barostat methods as these typically fix both temperature and pressure.

A wide range of different approaches can be designed to sample from the canonical distribution; each such formulation has certain desirable properties, but also certain limitations. For example it is well known that the dynamical response of a system simulated using Langevin dynamics will strongly depend on the friction coefficient. However, since Langevin dynamics relies on a strong assumption of scale separation, it is

possible that there is no choice of the friction that gives a satisfactory dynamical approximation^{30–33} (see also discussions on time scales associated with Langevin dynamics³³). In some cases a “gentle” form of Langevin dynamics is used where internal relaxation modes within the polymer are the dominant feature of interest, or else one uses Nosé–Hoover^{34,35} or stochastic velocity rescaling,^{36–38} both of which are in some sense “gentle” alternatives to Langevin dynamics.³⁹ Other approaches include the pairwise Nosé–Hoover thermostat⁴⁰ and multiparticle collision dynamics.⁴¹ However, we restrict our attention to methods that are derived directly as discretizations of stochastic differential equations. In case the goal is only to sample the equilibrium structures of the system under study (as often needed in protein modeling^{42,43}), one may use the coefficient of friction as a free parameter and optimize the choice to enhance the rate of convergence to equilibrium or increase the effective sample size⁴⁴ associated with a certain family of observables. Whereas standard molecular dynamics methods such as Langevin dynamics and its overdamped limit (Brownian dynamics) are appropriate for modeling systems in or near thermodynamic equilibrium, these methods do not take into account the possibility of an underlying flow, and are thus, unless modified, inappropriate for situations where the underlying flow of the system cannot be predicted beforehand (*e.g.*, when dealing with interfaces or nonuniform flow).

More generally, as we coarse-grain the system, the hydrodynamic transport properties become increasingly important, which calls for a formulation that preserves momentum and Galilean invariance. A method which addresses these issues is the dissipative particle dynamics (DPD) method of Hoogerbrugge and Koelman⁴⁵ which conserves momentum “exactly” (*i.e.*, up to rounding errors) at each step. We find in our studies that the recently proposed pairwise adaptive Langevin (PADL) method,⁴⁶ which mimics DPD in the stationary setting, is preferable to DPD in nonequilibrium applications, *e.g.*, for shear flows. The use of DPD or PADL addresses a further problem with Langevin dynamics, namely the nonphysical screening of hydrodynamic interactions observed for Langevin dynamics.⁴⁷

Once the formulation is chosen, typically a set of stochastic differential equations (SDEs), it is necessary to replace it for computation with a discretized formulation, by applying a specific numerical method. Particularly for large, computationally demanding applications, the simulation must proceed at the largest possible timestep in order to provide meaningful answers to the questions of interest. Since we must work within a fixed computational budget, one should ultimately compare different numerical methods in terms of accuracy of observables for fixed computational work. On the other hand, if, as here, we wish to separate the numerical error (due to discrete approximation) from the statistical error (due to collecting finite numbers of samples) it is better to analyze these two types of error separately. This is especially true in a metastable system, *i.e.*, one whose dynamics are limited by rare transitions, as for example protein folding or glassy system modeling,^{48,49} but generally speaking virtually any particle simulation will be



subject to the timestep issue when pushed to deliver results on laboratory-relevant timescales. Thus we are led to seek methods that are optimized to deliver the desired properties in the most cost-effective manner, and the large timestep needed means that discretization bias becomes relevant. For molecular and mesoscale modeling, the most fundamental type of error incurred is the error in averages. We say that a method is accurate for long term averaging if the distribution generated by the numerical method converges in the limit of large time and small timestep to a stationary distribution which is close, in the sense of distributions, to the corresponding stationary distribution of the SDE.^{50,51} The other major source of error in large scale simulations can be viewed as the error due to insufficient sampling of a stochastic quantity. Such error is always present but may be difficult to estimate, since in practice we do not necessarily know the underpinning distribution *a priori* nor the normalization constant needed to compute a robust average. To put this another way, if we remain for our entire simulation time exploring certain states of the molecule, it might be the consequence of a strong local confinement rather than an indication that these are always the most important group of states. Analyzing and comparing numerical methods for sampling thus requires balancing the issues of accuracy and sampling efficiency and it is crucial to realize that different methods may have very different levels of bias and rates of convergence which are highly dependent on the choice of timestep. In this article we explore these issues in detail for model systems, calculating first the bias and using this to select timestep to control the attainable accuracy, then (for the specific choice of timestep) assessing the rate of convergence to equilibrium average.

The key findings of this article are as follows. First, we compare the performance of Langevin dynamics, DPD, and PAdL for several benchmark calculations, in particular showing that the PAdL method provides higher accuracy (for given computational budget) than Langevin dynamics and DPD, in both equilibrium and nonequilibrium applications, with careful study of the trade-off between numerical bias and convergence rate in simulation. Second, we compare our numerical schemes with exact results regarding the relaxation behavior for a benchmark model, thus clarifying the performance of the methods in dynamics-oriented modeling applications. Third, we develop a careful procedure to quantify the sampling efficiency of various methods by comparing the effective sample size. A fourth advance in the current article is the demonstration that PAdL allows the recovery of accurate shear viscosity using larger shear rates than otherwise are possible using DPD (at identical timestep) while PAdL outperforms Langevin dynamics in terms of stability and accuracy at higher shear rates. Finally, we emphasize that this article provides specific details regarding implementation of all the various methods which are often lacking in the literature.

The rest of the article is organized as follows. In Section 2, we review a variety of numerical methods in polymer melts simulation. We describe, in Section 3, various physical quantities that are used to evaluate the simulation performance of

each method. Section 4 presents numerical experiments in both equilibrium and nonequilibrium cases, comparing the performance of numerous popular numerical methods in practical examples. Our findings are summarized in Section 5.

2 Numerical methods

In this section, we describe various numerical methods used to simulate many-particle systems. We are interested both in the choice of formulation of the equations of motion and in the consequent secondary choice of discretization method. In the literature one observes that virtually all the popular methods are of a relatively simple design and require typically a single evaluation of the forces of interaction at each timestep, the computational cost of the force evaluation being normally the unit of computational effort.

2.1 Langevin thermostat

Following the seminal work of Grest and Kremer,^{52,53} Langevin dynamics has been widely used in simulating Lennard-Jones systems including polymer chains and their melts and can be written as

$$\begin{aligned} d\mathbf{q}_i &= m^{-1}\mathbf{p}_i dt, \\ d\mathbf{p}_i &= \mathbf{F}_i(\mathbf{q}, t)dt - \gamma(\mathbf{p}_i - m\mathbf{u}_i)dt + \sigma m^{1/2}d\mathbf{W}_i, \end{aligned} \quad (1)$$

where \mathbf{q}_i and \mathbf{p}_i are d -dimensional vectors and respectively represent positions and absolute momenta of bead i with d being the underlying dimensionality of the physical space (typically $d = 3$), m denotes the mass of a particle, the force on particle i , $\mathbf{F}_i(\mathbf{q}, t)$, could in principle be both positions and time dependent, however, in equilibrium, $\mathbf{F}_i = -\nabla_{\mathbf{q}_i}U$ is the conservative force given in terms of a potential energy function $U = U(\mathbf{q})$, $d\mathbf{W}_i$ represents a dimensionless vector of d independent increments of Wiener processes with stochastic properties $\langle d\mathbf{W}_i(t) \rangle = 0$ and $\langle d\mathbf{W}_i(t)d\mathbf{W}_j(t') \rangle = \delta_{ij}\delta(t - t')1dt$, γ is the bead friction coefficient, which couples the beads weakly to a heat bath, $\mathbf{u}_i = \mathbf{u}(\mathbf{q}_i)$ denotes a macroscopic streaming velocity at position \mathbf{q}_i , and σ represents the strength of the random forces, satisfying the following fluctuation–dissipation relation:

$$\sigma^2 = 2\gamma k_B T. \quad (2)$$

It should be emphasized here that the damping term in (1) depends on the peculiar velocity, which is the difference between the absolute velocity $\mathbf{v}_i = \mathbf{p}_i/m$ and the streaming velocity field \mathbf{u} . That is, the thermostat acts only on the peculiar velocity, which is essential in nonequilibrium, for instance, when modeling shear flow. The traditional formulation of Langevin dynamics (*i.e.*, when $\mathbf{u} = 0$) does not take into account the underlying streaming velocity and thus is expected to fail when there exists a nonzero underlying streaming velocity, however, this feature is not always clearly stated.⁵⁴

2.1.1 Stochastic velocity Verlet (SVV). Due to its ease of implementation and its natural construction based on the popular Verlet method of molecular dynamics, the stochastic



velocity Verlet (SVV) method⁵⁵ is one of the most popular methods for Langevin dynamics. The equations are:

$$\begin{aligned}\mathbf{p}_i^{n+1/2} &= \mathbf{p}_i^n - \frac{h}{2}\nabla_{\mathbf{q}_i}U(\mathbf{q}^n) - \frac{h}{2}\gamma(\mathbf{p}_i^n - m\mathbf{u}_i) + \sqrt{\frac{hm}{2}}\sigma\mathbf{R}_i^n, \\ \mathbf{q}_i^{n+1} &= \mathbf{q}_i^n + hm^{-1}\mathbf{p}_i^{n+1/2}, \\ \mathbf{p}_i^{n+1} &= \mathbf{p}_i^{n+1/2} - \frac{h}{2}\nabla_{\mathbf{q}_i}U(\mathbf{q}^{n+1}) - \frac{h}{2}\gamma(\mathbf{p}_i^{n+1/2} - m\mathbf{u}_i) \\ &\quad + \sqrt{\frac{hm}{2}}\sigma\mathbf{R}_i^{n+1/2},\end{aligned}$$

where h is the integration timestep, \mathbf{R}_i^n and $\mathbf{R}_i^{n+1/2}$ are vectors of uncorrelated Gaussian white noise with zero mean and unit variance, resampled at each step.

2.1.2 The BAOAB method. Numerical integration methods for Langevin dynamics have been studied systematically in terms of the long term sampling performance in recent works of Leimkuhler and Matthews.^{50,56} Of note is the observation that a particular choice of splitting method, “BAOAB”, based on a Trotter factorization of the stochastic vector field of the system into exactly solvable subsystems, is far superior to alternative methods in terms of sampling configurational quantities. The BAOAB method relies on separating the vector field of the system:

$$\begin{aligned}d\begin{bmatrix} \mathbf{q}_i \\ \mathbf{p}_i \end{bmatrix} &= \underbrace{\begin{bmatrix} m^{-1}\mathbf{p}_i \\ \mathbf{0} \end{bmatrix}}_A dt + \underbrace{\begin{bmatrix} \mathbf{0} \\ -\nabla_{\mathbf{q}_i}U \end{bmatrix}}_B dt \\ &\quad + \underbrace{\begin{bmatrix} \mathbf{0} \\ -\gamma(\mathbf{p}_i - m\mathbf{u}_i)dt + \sigma m^{1/2}d\mathbf{W}_i \end{bmatrix}}_O,\end{aligned}\quad (3)$$

in such a way that each piece can be solved “exactly”. It is straightforward to solve the “A” and “B” pieces, respectively. Moreover, it is possible to derive the exact solution to the Ornstein–Uhlenbeck (“O”) part,

$$d\mathbf{p}_i = \gamma m\mathbf{u}_i dt - \gamma\mathbf{p}_i dt + \sigma m^{1/2}d\mathbf{W}_i, \quad (4)$$

as

$$\mathbf{p}_i(t) = m\mathbf{u}_i + (\mathbf{p}_i(0) - m\mathbf{u}_i)e^{-\gamma t} + \sigma\sqrt{\frac{m(1 - e^{-2\gamma t})}{2\gamma}}\mathbf{R}_i, \quad (5)$$

where \mathbf{R}_i is a vector of independent and identically distributed (i.i.d.) standard normal random variables. The BAOAB method then can be defined as

$$e^{h\mathcal{L}_{\text{BAOAB}}} = e^{(h/2)\mathcal{L}_B}e^{(h/2)\mathcal{L}_A}e^{h\mathcal{L}_O}e^{(h/2)\mathcal{L}_A}e^{(h/2)\mathcal{L}_B}, \quad (6)$$

where $\exp(h\mathcal{L}_f)$ denotes the phase space propagator associated with the corresponding vector field f . The integration steps

of the BAOAB method, modified to include the streaming velocity, reads:

$$\begin{aligned}\mathbf{p}_i^{n+1/2} &= \mathbf{p}_i^n - (h/2)\nabla_{\mathbf{q}_i}U(\mathbf{q}^n), \\ \mathbf{q}_i^{n+1/2} &= \mathbf{q}_i^n + (h/2)m^{-1}\mathbf{p}_i^{n+1/2}, \\ \tilde{\mathbf{p}}_i^{n+1/2} &= m\mathbf{u}_i + (\mathbf{p}_i^{n+1/2} - m\mathbf{u}_i)e^{-\gamma h} + \sqrt{mk_B T(1 - e^{-2\gamma h})}\mathbf{R}_i^n, \\ \mathbf{q}_i^{n+1} &= \mathbf{q}_i^{n+1/2} + (h/2)m^{-1}\tilde{\mathbf{p}}_i^{n+1/2}, \\ \mathbf{p}_i^{n+1} &= \tilde{\mathbf{p}}_i^{n+1/2} - (h/2)\nabla_{\mathbf{q}_i}U(\mathbf{q}^{n+1}).\end{aligned}$$

It should be noted that only one force calculation is required at each step for BAOAB (*i.e.*, the force computed at the end of each step will be reused at the beginning of the next step), the same as for alternative schemes, including the SVV method.

2.2 Dissipative particle dynamics (DPD) thermostat

Momentum conservation is an essential property required to correctly capture hydrodynamic interactions. However, the momentum is not conserved in Langevin dynamics due to the fact that the thermostat (*i.e.*, the dissipative and random forces) is not pairwise. Analogous to Langevin dynamics, the dissipative particle dynamics (DPD) method^{45,57,58} is a momentum-conserving thermostat which has been proposed to simulate complex hydrodynamic behavior. Unlike Langevin dynamics, the dissipative force in DPD is dependent of relative velocities and both the dissipative and random forces are pairwise, ensuring the momentum conservation. It should be noted that DPD has been used primarily as a mesoscale coarse-graining technique, where each DPD particle represents a blob of molecules, however, it has also been used in simulating polymer melts, in which case each DPD particle corresponds to one bead (*e.g.*, see ref. 59–61).

The equations of motion of the DPD system can be written as

$$\begin{aligned}d\mathbf{q}_i &= m^{-1}\mathbf{p}_i dt, \\ d\mathbf{p}_i &= \sum_{j \neq i} \left[\mathbf{F}_{ij}^C(r_{ij})dt - \gamma\omega^D(r_{ij})(\mathbf{e}_{ij} \cdot \mathbf{v}_{ij})\mathbf{e}_{ij}dt + \sigma\omega^R(r_{ij})\mathbf{e}_{ij}d\mathbf{W}_{ij} \right],\end{aligned}$$

where $r_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|$ is the distance between particles i and j with $\mathbf{e}_{ij} = (\mathbf{q}_i - \mathbf{q}_j)/r_{ij}$ being the unit vector in the associated direction, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ is the relative velocity, $\mathbf{F}_{ij}^C(r_{ij})$ denotes the conservative force derived from the corresponding pair potential energy $U(r_{ij})$, and $d\mathbf{W}_{ij} = d\mathbf{W}_{ji}$ are independent increments of Wiener processes with mean zero and variance dt . In addition to the relation in (2), the two weight functions have to be related by $\omega^D(r_{ij}) = [\omega^R(r_{ij})]^2$ in order for the system to sample the canonical ensemble.

We have observed⁶² that standard DPD methods perform similarly in all the quantities that we have tested. Therefore, following ref. 46, Shardlow’s S1 splitting method (*i.e.*, the DPD-S1 scheme)⁶³ was used to represent the standard DPD formulation. As in Langevin dynamics, we can similarly define



the DPD-S1 (OBAB) method as

$$e^{h\mathcal{L}_{\text{DPD-S1}}} = e^{h\mathcal{L}_O} e^{(h/2)\mathcal{L}_B} e^{h\mathcal{L}_A} e^{(h/2)\mathcal{L}_B}, \quad (7)$$

where one should note that the “O” part is further split into interacting pairs and then each pair is solved by using the method of Brünger, Brooks, and Karplus (BBK)⁶⁴ (the detailed integration steps of the DPD-S1 scheme can be found in Appendix A).

Due to the fact that the dissipative force depends on relative velocities, DPD is Galilean-invariant, which makes it a profile-unbiased thermostat (PUT)^{65,66} by construction and an ideal thermostat for nonequilibrium molecular dynamics (NEMD).⁵⁴ The PUT allows the simulation itself to define the local streaming velocity (for more details, see ref. 65–67) and thus there is no need to additionally subtract the underlying streaming velocity in nonequilibrium applications.

2.3 Pairwise adaptive Langevin (PAdL) thermostat

Inspired by recent developments in adaptive thermostats,^{68–70} the pairwise adaptive Langevin (PAdL) thermostat, which can be viewed as “adaptive DPD”, has been proposed by Leimkuhler and Shang,⁴⁶ see also more discussions therein. It has been observed that PAdL is able to correct for thermodynamic observables while mimicking the dynamical properties of DPD.

The equations of motion of the momentum-conserving PAdL thermostat are given by

$$d\mathbf{q}_i = m^{-1}\mathbf{p}_i dt,$$

$$d\mathbf{p}_i = \sum_{j \neq i} \left[\mathbf{F}_{ij}^C(r_{ij}) dt - \xi \omega^D(r_{ij}) (\mathbf{e}_{ij} \cdot \mathbf{v}_{ij}) \mathbf{e}_{ij} dt + \sigma \omega^R(r_{ij}) \mathbf{e}_{ij} d\mathbf{W}_{ij} \right],$$

$$d\xi = G(\mathbf{q}, \mathbf{p}) dt,$$

where ξ is an auxiliary dynamical friction variable, σ a constant amplitude as in Langevin dynamics, and $G(\mathbf{q}, \mathbf{p})$ denotes the accumulated deviation of the instantaneous temperature away from the target temperature

$$G(\mathbf{q}, \mathbf{p}) = \frac{1}{\mu} \sum_i \sum_{j>i} \omega^D(r_{ij}) \left[(\mathbf{v}_{ij} \cdot \mathbf{e}_{ij})^2 - 2k_B T/m \right], \quad (8)$$

where μ is a coupling parameter (an inverse surface mass density) which is referred to as the “thermal mass”. It can be shown that, in equilibrium, the PAdL system preserves the momentum-constrained canonical ensemble with a modified density

$$\begin{aligned} \tilde{\rho}_\beta(\mathbf{q}, \mathbf{p}, \xi) &= \frac{1}{Z} \exp \left[-\beta H(\mathbf{q}, \mathbf{p}) - \frac{\beta\mu}{2} (\xi - \gamma)^2 \right] \\ &\times \delta \left(\sum_i p_i^x - \pi_x \right) \delta \left(\sum_i p_i^y - \pi_y \right) \delta \left(\sum_i p_i^z - \pi_z \right), \end{aligned} \quad (9)$$

where γ is the friction coefficient as it satisfies the fluctuation–dissipation relation (2), and $\pi = (\pi_x, \pi_y, \pi_z)$ is the linear momentum vector. Additional modifications should be included if the angular momentum is also conserved.

According to the invariant distribution (9), the auxiliary variable ξ is Gaussian distributed with mean γ and variance $(\beta\mu)^{-1}$. That is, the auxiliary variable will fluctuate around its mean value during simulation and moreover we can vary the value of the friction in order to recover the dynamics of DPD in a wide range of friction regimes. Therefore, the PAdL thermostat can be viewed as the standard DPD system with an adaptive friction coefficient (*i.e.*, an adaptive DPD thermostat). Furthermore, we point that the PAdL thermostat inherits key properties of DPD (such as Galilean invariance and momentum conservation) required for consistent hydrodynamics. Note also that the PAdL thermostat would effectively reduce to the standard DPD formulation in the large thermal mass limit (*i.e.*, $\mu \rightarrow \infty$).

The splitting method of PAdL proposed in ref. 46 has been adopted in this article:

$$e^{h\hat{\mathcal{L}}_{\text{PAdL}}} = e^{(h/2)\mathcal{L}_A} e^{(h/2)\mathcal{L}_B} e^{(h/2)\mathcal{L}_O} e^{h\mathcal{L}_D} e^{(h/2)\mathcal{L}_O} e^{(h/2)\mathcal{L}_B} e^{(h/2)\mathcal{L}_A},$$

where the “O” part is again further split into interacting pairs but each pair is solved exactly, and “D” represents the additional Nosé–Hoover part (the detailed integration steps of the PAdL method can also be found in Appendix A). It is worth mentioning that the computational costs per timestep of all the methods examined in this article are very similar. With the help of computation-saving devices such as Verlet neighbor lists,⁷¹ the computational effort of all methods under study scales with the number of particles, and both DPD and PAdL are only slightly more expensive than Langevin integrators.

3 Quantifiers for simulation performance

In this section, we briefly outline the quantities we will compute in simulation and use to compare the performance of different simulation schemes. These divide into observables for equilibrium and nonequilibrium sampling and the rate of convergence to equilibrium.

3.1 Summed autocorrelation count (SAC)

As in Markov chain Monte Carlo methods, we are interested in accurately and efficiently estimating the expected value of some physical observable of interest $f(x)$, *i.e.*,

$$\mathbb{E}[f] = \langle f \rangle = \int_{\Omega_x} f(x) \rho(x) dx, \quad (10)$$

by averaging a time series of (typically correlated) N_s samples

$$\bar{f} = \frac{1}{N_s} \sum_{i=1}^{N_s} f(x_i) \quad (11)$$

for large N_s , where x_i denotes the phase space configuration at time t_i . Denoting the variance of f with respect to the probability density function ρ by σ_f^2 , which is independent of particular sampling methods, it can be shown⁷² that the variance of the



estimator \bar{f} of the mean is

$$\bar{\sigma}_\rho^2(\bar{f}) = \frac{\tau\sigma_\rho^2}{N_s}, \quad (12)$$

where τ indicates a quantity that we refer to as the “summed autocorrelation count” (SAC),[†] for correlated samples, typically estimated as $\tau \equiv \tau(k_{\max})$ for some finite $k_{\max} < N_s$ by the “running” autocorrelation count estimator[‡]

$$\tau(k) = 1 + 2 \sum_{j=1}^k \left(1 - \frac{j}{N_s}\right) \frac{C(j)}{C(0)}, \quad (13)$$

where

$$C(j) = \frac{1}{N_s - j} \sum_{i=1}^{N_s-j} (f(x_i) - \bar{f})(f(x_{i+j}) - \bar{f}), \quad (14)$$

is the unnormalized autocorrelation function (or auto covariance) of f . If the samples are uncorrelated, $\tau = 1$, in which case the variance of the estimator \bar{f} would simply be σ_ρ^2/N_s . Note that the variance of f (*i.e.*, σ_ρ^2) is a special case of the autocorrelation (14), *i.e.*, $\sigma_\rho^2 = C(0)$. The running $\tau(k)$ starts at unity for $k = 0$, vanishes exactly for $k = N_s - 1$, and its value and variance go through a maximum for intermediate k . In our simulations, we found that it was unclear how to properly determine the k_{\max} . Therefore, we instead suggest to approximate the SAC based on a weighted sum fitting⁷⁴ of the normalized autocorrelation function, whose argument is scaled to physical time, t , for convenience:

$$\frac{C(t)}{C(0)} = (1 - c)e^{-t/\lambda_1} + c \left(\cos(\omega t) + \frac{\sin(\omega t)}{w\lambda_2} \right) e^{-t/\lambda_2}, \quad (15)$$

where $c \in [0,1]$ is dimensionless, $\lambda_1, \lambda_2 > 0$ are time constants, and ω is a frequency. Integrating (15) from zero to infinity yields

$$I \equiv \int_0^{+\infty} \frac{C(t)}{C(0)} dt = (1 - c)\lambda_1 + \frac{2c\lambda_2}{1 + (\lambda_2\omega)^2}, \quad (16)$$

based on which the SAC can be approximated as

$$\tau \approx \frac{2I}{h} - 1, \quad (17)$$

where h is the stepsize associated with the numerical method. The choice of the functional form (15) is somewhat arbitrary, as it simply corresponds to the c -weighted superposition of solutions of a damped harmonic oscillator and a monoexponential relaxation process, but we observed good agreement with the

[†] Note that the SAC is often referred to as the “integrated autocorrelation time”^{44,72,73} in the computational statistics literature. The problem with using such a term here relates to the fact that the time is a well-defined physical quantity whereas the formula quantifies a number of steps of an iterative procedure. Since we are mostly interested in how quickly the samples decorrelate in terms of the number of steps, we use the word “count” in order to avoid confusion with “physical time”.

[‡] It should be noted that MATLAB’s “autocorr” function, despite its name, does not calculate the autocorrelation function (14), however, $1 + \tau(k_{\max})$ is just two times the first $1 + k_{\max}$ “autocorr” lags in MATLAB.

actual autocorrelation function behavior in our simulation experiments (see the next section).

It should be noted that the SAC is closely related to the statistical error bar, *i.e.*, the statistical error bar of the estimated mean is the standard deviation of the time series divided by the “effective sample size” defined as N_s/τ . Thus, the SAC is an estimate of the number of iterations, on average, for an independent sample to be drawn, given a correlated chain. Therefore, the SAC directly measures the efficiency of the sampling—a lower value of SAC corresponds to a larger effective sample size, *i.e.*, a more efficient sampling.

3.2 Configurational temperature

Since we are mostly interested in configurational sampling, in calculating the SAC we choose for f the configurational temperature,^{75–79}

$$k_B f_T = \frac{\langle \nabla U(\mathbf{q}) \cdot \nabla U(\mathbf{q}) \rangle}{\langle \nabla^2 U(\mathbf{q}) \rangle}, \quad (18)$$

an observable function solely depending on positions whose average in the canonical ensemble, as the kinetic temperature, is precisely the target temperature:

$$T = \langle f_T \rangle, \quad (19)$$

where ∇U and $\nabla^2 U$ respectively denote the gradient and Laplacian of the potential energy U in the configurational phase space (further discussions in ref. 62). The corresponding unnormalized correlation function and running SAC are denoted by $C_T(k)$ and $\tau_T(k)$, respectively.

3.3 Polymer conformation

A multibead nonlinear spring model was employed to simulate a polymer melt as described in detail in Section 4.1. Denote the coordinate of the j -th bead in chain α as $\mathbf{q}_j^{(\alpha)}$. The end-to-end vector of chain α is then given by $\mathbf{R}_{ee}^{(\alpha)} = \mathbf{q}_N^{(\alpha)} - \mathbf{q}_1^{(\alpha)}$. It should be emphasized here that in taking differences one has to respect the periodic boundary conditions (or to simply unfold all polymer contours before applying the above definitions if they are not already kept unfolded within the code).

Correlation functions, which characterize the relevant dynamical properties, are often studied in molecular dynamics. Of particular interest in polymer melts is the orientational autocorrelation function (OAF) of the end-to-end vector of polymer chains, which characterizes the relaxation of the polymer chains and is evaluated by choosing for f the end-to-end vector $\mathbf{R}_{ee}^{(\alpha)}(t)$ of chain α , while all M chains contribute to $C_{ee}(t)$ as individual samples. For this vector-valued f the product in eqn (14) is a scalar product. Due to head–tail symmetry $\langle \mathbf{R}_{ee}^{(\alpha)} \rangle = 0$ and thus $\bar{f} = 0$ in that case.

3.4 Shear viscosity

A common approach to generate a simple shear flow in nonequilibrium molecular dynamics is to apply the well-known Lees–Edwards boundary conditions (LEBC),⁸⁰ where, as in normal periodic boundary conditions (PBC), the primary cubic



box remains centered at the origin, however, a uniform shear velocity profile is expected⁶⁶

$$\mathbf{u}_i = \dot{\gamma}(\mathbf{q}_i \cdot \mathbf{e}^y) \mathbf{e}^x = \boldsymbol{\kappa} \cdot \mathbf{q}_i, \quad \boldsymbol{\kappa} = \dot{\gamma} \mathbf{e}^x \otimes \mathbf{e}^y \quad (20)$$

where \mathbf{e}^x and \mathbf{e}^y respectively denote the unit vector in the x - and y -direction, $\boldsymbol{\kappa}$ is the transposed velocity gradient tensor, \otimes represents the dyadic product of two vectors, and $\dot{\gamma}$ is the shear rate defined as $\dot{\gamma} = du^x/dy$, where u^x is the macroscopic velocity in the x -direction. It is worth mentioning that while LEBC is typically applied only in the x -direction, the other directions (y and z) remain with PBC. It is nontrivial to implement LEBC in pairwise thermostats due to the position-dependence on both dissipative and random forces, this issue has been discussed in ref. 46.

The Irving–Kirkwood stress tensor⁸¹ subject to LEBC can be written as

$$\boldsymbol{\sigma} = -\frac{1}{V} \left(\sum_i m(\mathbf{v}_i - \mathbf{u}_i) \otimes (\mathbf{v}_i - \mathbf{u}_i) + \sum_i \sum_{j>i} \mathbf{q}_{ij} \otimes \mathbf{F}_{ij} \right), \quad (21)$$

where V is the volume of the simulation box, and \mathbf{u}_i is the streaming velocity (20) corresponding to the location of particle i . Only the conservative force should be included for \mathbf{F}_{ij} in Langevin dynamics since both the dissipative and random forces are averaged out, whereas all three components of the force should be accounted for pairwise thermostats. The generally non-Newtonian shear viscosity is extracted at finite rates as

$$\eta = \frac{\langle \sigma_{xy} \rangle}{\dot{\gamma}}, \quad (22)$$

where σ_{xy} denotes the shear stress, which is the off-diagonal xy -component of the symmetric stress tensor $\boldsymbol{\sigma}$ (21). While employing (22) the zero shear viscosity $\eta_0 = \lim_{\dot{\gamma} \rightarrow 0} \eta$ can be obtained by extrapolation, it is worth mentioning that η_0 can be alternatively calculated by integrating the stress–stress auto-correlation function (*i.e.*, the Green–Kubo formulas^{82,83}). However, it is well documented that those equilibrium approaches are subject to significant statistical error and thus not preferred in practice (see a detailed discussion on extracting transport coefficients by various approaches in ref. 84).

3.5 Flow alignment angle

As a nontrivial application to demonstrate the performance of sampling schemes in the nonequilibrium context, we study the flow alignment of the polymer segments to the imposed flow as a function of the shear rate imposed using LEBC in manner described in Section 3.4, thus we calculate the flow alignment angle⁸⁵ as follows:

$$\chi = \frac{\pi}{4} - \frac{1}{2} \arctan \left(\frac{\langle b_x^2 - b_y^2 \rangle}{2 \langle b_x b_y \rangle} \right), \quad (23)$$

where b_x and b_y respectively represent the x - and y -component of a normalized bond vector $\mathbf{b} = b_x \mathbf{e}^x + b_y \mathbf{e}^y + b_z \mathbf{e}^z$, and the average is taken over all bonds.

4 Numerical experiments

In this section, we conduct systematic numerical experiments to compare the performance of various methods introduced in Section 2 in polymer melts simulations.

4.1 Simulation details

A popular bead-spring model originally proposed by Kremer and Grest^{52,53} is used in our simulations. The system is composed of M identical linear chains with N beads each in a cubic box with periodic boundary conditions.⁸⁶ The total number of beads is $N_t = MN$ in that case. Excluded volume interactions between all N_t beads are included *via* a truncated Lennard-Jones potential:

$$U_{\text{LJ}}(r_{ij}) = \begin{cases} 4\epsilon \left[\left(\frac{r_0}{r_{ij}} \right)^{12} - \left(\frac{r_0}{r_{ij}} \right)^6 - \left(\frac{r_0}{r_c} \right)^{12} + \left(\frac{r_0}{r_c} \right)^6 \right], & r_{ij} < r_c; \\ 0, & r_{ij} \geq r_c, \end{cases}$$

where $r_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|$ denotes the distance between two beads i and j , ϵ and r_0 are two constants that set the energy and length scales of the beads, respectively, in reduced Lennard-Jones units, and r_c is the cutoff radius typically chosen as $r_c = 2^{1/6} r_0$ such that only the repulsive part of the potential is considered. This potential is also known as the Weeks–Chandler–Andersen potential.⁸⁷

Adjacent N beads along the same polymer interact, in addition, *via* the finitely extensible nonlinear elastic (FENE) potential:

$$U_{\text{FENE}}(r_{ij}) = \begin{cases} -\frac{1}{2} k R_{\text{max}}^2 \ln \left[1 - (r_{ij}/R_{\text{max}})^2 \right], & r_{ij} < R_{\text{max}}; \\ \infty, & r_{ij} \geq R_{\text{max}}, \end{cases}$$

where $k = 30\epsilon/r_0^2$ represents the spring coefficient and $R_{\text{max}} = 1.5r_0$ determines the maximum length of a bond. This choice of parameters ensures that chains do not cross each other and it allows for a reasonable large integration timestep.⁵³ The system is thermostated as described in Section 2.

Overall, the total potential energy of the system is defined as

$$U(\mathbf{q}) = \sum_{i=1}^{N_t-1} \sum_{j=i+1}^{N_t} [U_{\text{LJ}}(r_{ij}) + U_{\text{FENE}}(r_{ij})], \quad (24)$$

and the total potential of a bond, $U_{\text{LJ}}(r_{ij}) + U_{\text{FENE}}(r_{ij})$ gives rise to a mean bond length $\langle r_{ij} \rangle \approx 0.97r_0$ between adjacent beads i and j for $\epsilon = k_B T = 1$.

A simple and popular choice of the weight function as in ref. 58 was adopted in this article:

$$\omega_{ij}^{\text{R}} = \omega^{\text{R}}(r_{ij}) = \begin{cases} 1 - \frac{r_{ij}}{r_c}, & r_{ij} < r_c; \\ 0, & r_{ij} \geq r_c. \end{cases} \quad (25)$$

A system (bead number density $\rho_d = 0.84$) consisting of $M = 30$ identical linear chains with $N = 20$ beads (unit mass m) on each chain was simulated, where the following parameter set was used: $k_B = \epsilon = r_0 = m = 1$ (defining reduced units) at $T = 1$ in reduced units. The thermal mass in PADL was initially



chosen as $\mu = 10$. It should be noted that the simulated system is usually referred to as “unentangled” since $N \leq 85$.⁸⁵ Pre-equilibrated initial configurations were obtained using an existing hybrid approach:⁸⁸ Initially, a mixture of phantom and excluded volume FENE chains were placed randomly into the simulation box at a density that exceeds the target density. A subsequent molecular dynamics algorithm with integration time step, force shape and force strength control was used to achieve a prescribed minimum distance (here 0.9) between all pairs of particles, while attempting to maintain local and global characteristics (such as the form factor) of the chain conformations. During this process the most inefficient chains were removed from the system, until the target density was reached. The initial momenta were independent and identically distributed (i.i.d.) normal random variables with mean zero and variance $k_B T$. Unless otherwise stated, the system was simulated for 1000 reduced time units in each case but only the last 80% of the snapshots were collected to calculate various quantities described in the preceding section.

4.2 Equilibrium

As a verification of our equilibrium simulations, we first investigate the structural quantities obtained by using the SVV, BAOAB, DPD, and PAdL methods introduced in Section 2. Among all the methods, at a stepsize of $h = 0.01$, we obtain an (time and chain) averaged squared end-to-end distance of $\langle R_{ee}^2 \rangle = 29.46$ and an averaged squared radius of gyration⁵³ of $\langle R_g^2 \rangle = 4.87$ in the low friction regime of $\gamma = 0.5$, in perfect agreement with the results of Kremer and Grest.⁵³

Fig. 1 compares the configurational temperature (19) control for a variety of methods with a range of friction coefficients. Note that SVV and BAOAB are two different splitting methods of Langevin dynamics. However, it can be clearly seen that, while maintaining a similar accuracy control of the configurational

temperature, the BAOAB method allows the use of much larger (at least doubled) stepsizes compared to the SVV method, especially in the large friction limit ($\gamma = 40.5$), where a super-convergence (*i.e.*, a fourth order convergence, indicated by the dashed black line in the figure, to the invariant distribution) result was observed (as in ref. 50 and 69). All the other methods tested show second order convergence according to the dashed order line. This again illustrates the importance of optimal design of numerical methods. It is also interesting to note that the relative error slightly rises as we increase the friction coefficient for the SVV method whereas, in the BAOAB method, the relative error decreases.

While the relative error of both SVV and BAOAB methods depends on the friction coefficients in Langevin dynamics, the two pairwise thermostats (*i.e.*, DPD and PAdL) appear to show little dependence on the friction coefficients. Although it seems that the DPD method is as accurate as SVV, the PAdL method is superior to both of them (even slightly better than BAOAB at low friction, *i.e.*, $\gamma = 0.5$).

The accuracy and rates of convergence for each method (and for each observable) depend in a nontrivial way on stepsize and so we cannot expect to use the same stepsize for different numerical integrators. In performing comparisons, it is crucial to develop a careful procedure to quantify sampling convergence in relation to the accuracy desired. In our studies we use a fixed accuracy threshold to select the stepsize for each method (in equilibrium) and then, for this choice of stepsize, which will be different for each method, we use the configurational temperature SAC (see Section 3.1) to estimate the convergence rate. The detailed protocol is as follows:

1. Choose a suitable observable (for instance, the configurational temperature throughout this article);
2. Determine the stepsize, h , for each method by requiring an identical accuracy of the sample mean $\langle f_T \rangle$ (for instance,

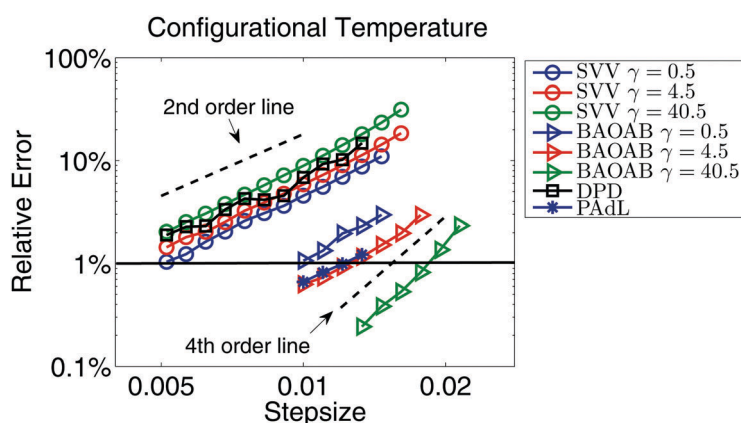


Fig. 1 Double logarithmic plot of the relative error, *i.e.*, the ratio between the absolute error of configurational temperature f_T (18) and the preset temperature T , against stepsize by using various numerical methods introduced in Section 2 with a variety of friction coefficients in a standard setting of polymer melts as described in Section 4.1. Note that the relative error of DPD and PAdL appears to show little dependence on the friction coefficients, thus the result of $\gamma = 0.5$ only is shown. The system was simulated for 1000 reduced time units in each case but only the last 80% of the snapshots were collected to calculate the static quantity f_T . Five different runs were averaged to further reduce the sampling errors. The stepsizes tested began at $h = 0.005$ and were increased incrementally by 10% until all methods became unstable. The horizontal solid black line indicates 1% relative error in sample mean (11) accuracy of configurational temperature, based on which the stepsizes for each method were chosen in equilibrium simulations, unless otherwise stated. Dashed black lines represent the second and fourth order convergence to the invariant distribution.



1% relative error, marked as the horizontal solid black line, as shown in Fig. 1);

3. The number of samples, N_s , for each method is subsequently specified as the total simulation time is kept fixed;

4. Approximate the SAC *via* eqn (16), which is based on a weighted sum fitting of the normalized autocorrelation function.

5. Calculate the effective sample size, N_s/τ_T , for each method, which characterize the sampling efficiency.

In order to measure the sampling efficiency of the various methods (three different values of the thermal mass, μ , of PAdL are included), we plot the normalized configurational temperature autocorrelation function (CTAF) and its corresponding fitted curve based on the weighted sum (15) in Fig. 2. The stepsize, corresponding to the $\sim 1\%$ relative error case, for each method was determined with the help of the horizontal solid black line in Fig. 1. As can be seen from the normalized autocorrelation functions in Fig. 2, the samples of various methods decorrelate very differently. It is interesting to note that while both Langevin dynamics and DPD exhibit “monotonic-like” decays, PAdL (with a wide range of the thermal mass) oscillates, which we believe is due to the presence of the additional Nosé–Hoover control.

As can be seen from Table 1, the SAC (τ_T) of DPD is significantly larger than alternatives. Moreover, the SAC of BAOAB is about half of that of SVV, which is due to the fact that BAOAB can use about double the stepsize of SVV and thus only about half as many samples are needed to achieve a similar sample mean accuracy of the configurational temperature. Although the SAC of PAdL with $\mu = 10$ is already smaller

Table 1 Comparisons of the sampling efficiency of various numerical methods quantified by the “effective sample size”, N_s/τ_T , in the low friction regime of $\gamma = 0.5$ with similar sample mean ($\langle f_T \rangle$) accuracy of $\sim 1\%$ relative error in configurational temperature. For all entries the total simulation time of collecting samples was $N_s h \approx 800$, where N_s and h respectively represent the number of samples and the stepsize, and I denotes the integrated normalized autocorrelation function (16) of f_T . The DPD method was computed by using Shardlow’s splitting method (*i.e.*, the DPD-S1 scheme).⁶³ The simulation details of the table are the same as in Fig. 1

| Method | h | N_s | $\langle f_T \rangle$ | I | τ_T | N_s/τ_T |
|------------------|-------|---------|-----------------------|--------|----------|--------------|
| SVV | 0.005 | 160 001 | 1.0105 | 0.4412 | 175.5 | 911.7 |
| BAOAB | 0.01 | 80 001 | 1.0134 | 0.4863 | 96.3 | 830.7 |
| DPD | 0.004 | 200 001 | 1.0093 | 1.1270 | 562.5 | 355.6 |
| PAdL $\mu = 10$ | 0.012 | 66 668 | 0.9903 | 0.2703 | 44.1 | 1511.7 |
| PAdL $\mu = 1$ | 0.012 | 66 668 | 0.9902 | 0.0959 | 15.0 | 4444.5 |
| PAdL $\mu = 0.1$ | 0.012 | 66 668 | 0.9902 | 0.0249 | 3.2 | 20833.8 |

than either Langevin dynamics or DPD, the SAC of PAdL could be decreased further by lowering the value of the thermal mass μ . This should not come as a great surprise since μ determines how strongly the negative feedback loops (8) couple with the physical system. Therefore, the PAdL method has the ability of controlling the sampling efficiency while others not.

Since different stepsizes, which result in different sample counts (N_s) when the total simulation time is fixed, were used for different methods in order to achieve a similar sample mean accuracy, one should further compare the sampling efficiency by computing the “effective sample size” (N_s/τ_T) instead of just the SAC (τ_T), which corresponds to cases where N_s is identical in each method. One can see from Table 1 that

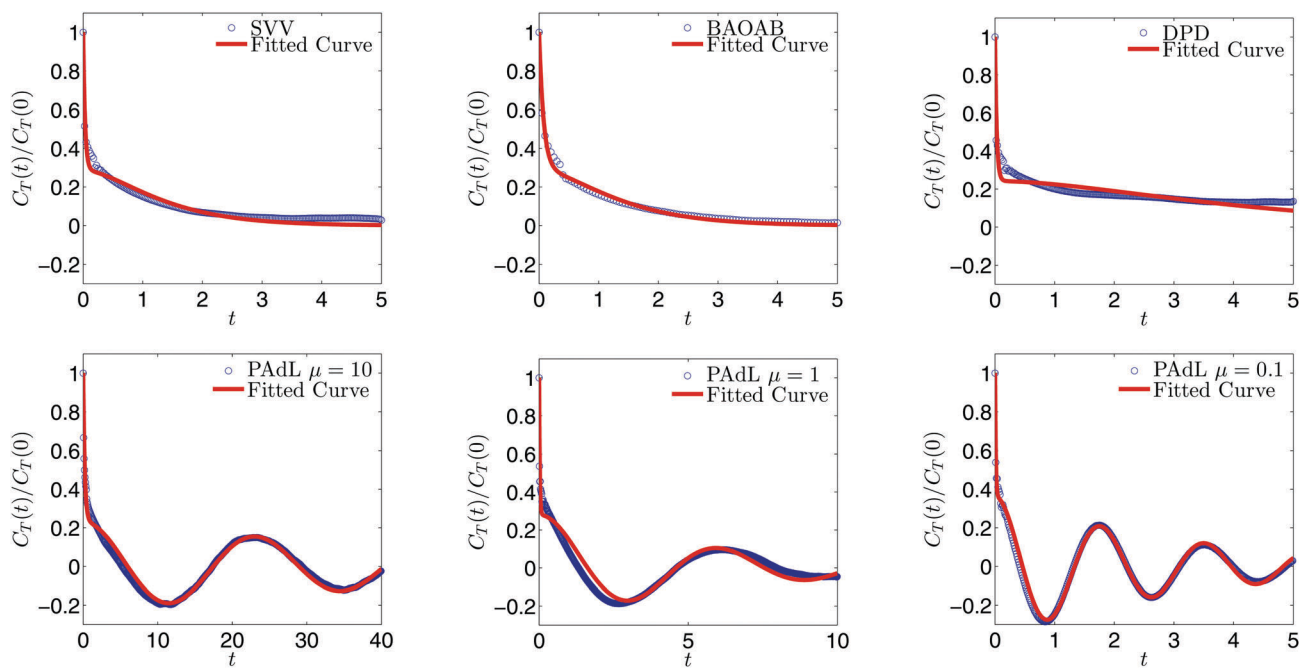


Fig. 2 Weighted sum fittings, employing (15), of the normalized configurational temperature autocorrelation function (14) by using various methods with fixed stepsizes shown in Table 1 in the low friction regime of $\gamma = 0.5$ with sample mean accuracy of $\sim 1\%$ relative error in configurational temperature. Note that the horizontal axis is scaled to the reduced physical time, rather than the number of steps, for convenience.



the effective sample sizes of SVV and BAOAB are very close to each other since they are just two different splitting methods of the same stochastic (Langevin) dynamics. While the effective sample size of DPD is roughly half that of Langevin dynamics (either SVV or BAOAB), PADL with $\mu = 10$ is over 50% larger than the latter. Further reducing the value of the thermal mass in PADL leads to an even more efficient sampling than alternatives: the effective sample size of PADL with $\mu = 1$ is more than four times that of DPD and Langevin dynamics; with $\mu = 0.1$, this increases to a factor of 20. There is a limit to how much μ can be reduced, however, without introducing numerical instability and thus requiring a smaller timestep. Note that although the thermal mass μ in PADL has a strong influence on the sampling efficiency, it appears that the sampling accuracy depends little on it (for instance, the long term behavior of PADL with a wide range of the thermal mass μ is almost indistinguishable in Fig. 1). Therefore, unless otherwise stated, $\mu = 0.1$ will be used in subsequent comparisons.

The characterization of the relaxation of polymer chains in a melt corresponding to different dynamics was compared and plotted in Fig. 3. In particular, we measured the orientational autocorrelation function (OAF) of the end-to-end vector of polymer chains defined in Section 3.3. It is believed that, for such a dense system, the long-time diffusion depends only on the interactions between beads and does not arise from the associated thermostat.⁵³ Therefore, the reference decay was calculated by using Hamiltonian dynamics (*i.e.*, switching off the thermostat, $\gamma = 0$). Since the OAFs obtained by SVV and BAOAB are almost indistinguishable, the SVV method was used for Langevin dynamics. (Note that in what follows, unless otherwise stated, Langevin dynamics was calculated by using the benchmark SVV method.)

It can be seen from Fig. 3 (left) that the OAF of Langevin dynamics depends strongly on the friction coefficient. To be more precise, the OAF starts to (significantly) deviate from the reference decay as we increase the friction coefficient.

Although a relative small friction (*i.e.*, $\gamma = 0.5$ in this parameter setting) was suggested in ref. 53 to not only minimize the effects of the Langevin thermostat but also to be large enough to stabilize the system in the long time limit, visible discrepancies were still observed in the case of $\gamma = 0.5$ in our numerical experiments. In stark contrast, the OAFs of the PADL method in a wide range of the friction coefficients are almost indistinguishable from the reference decay as shown in Fig. 3 (right). Very similar behavior was also observed in the DPD method, which implies that the projection of the interactions of both the dissipative and random forces (*i.e.*, the thermostat) on to the line of centers (and thus the conservation of the momentum) may have played a role in preserving the correct relaxation behavior in the case of the pairwise thermostats. Since a relatively small friction has been widely used in the literature of polymer melts, in what follows we restrict our attention to comparing various methods in the low friction regime of $\gamma = 0.5$.

4.3 Nonequilibrium

The stepsize for each method was chosen according to certain sample mean accuracy threshold (*e.g.*, $\approx 1\%$ relative error in configurational temperature) when examining the sampling efficiency in the previous subsection. However, we are more interested in investigating the stability issues in nonequilibrium simulations. Thus in what follows we fix the stepsize of $h = 0.01$, which is close to the stability threshold, for all methods.

The shear viscosity was extracted by using the formula (22) outlined in Section 3.4 and plotted in Fig. 4. All methods appear to show similar behavior except that the range of shear rates in DPD is greatly limited (that is, the largest usable shear rate in DPD is $\dot{\gamma} = 0.08$, compared with $\dot{\gamma} = 1.2$ in both Langevin and PADL). The error bars were indeed included, however they were relatively very small (particularly in the large shear rate regime) and thus not visible.

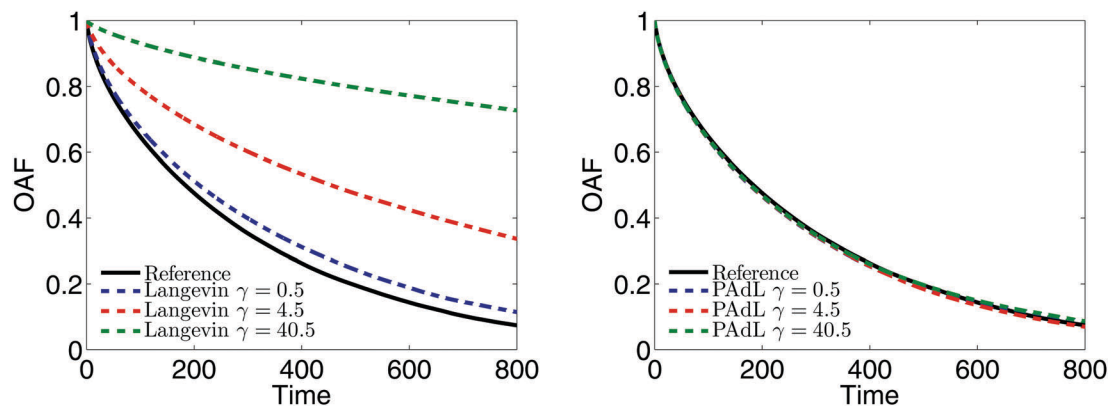


Fig. 3 Comparisons of the normalized orientational autocorrelation function (OAF), $C_{ee}(t)/C_{ee}(0)$, of the end-to-end vector of polymer chains in a melt between Langevin dynamics (left) and the PADL method (right) with three different values of the friction coefficient. Note that PADL and DPD exhibit indistinguishable behavior and thus only the result of the former was shown. The same stepsize of $h = 0.01$ was used for all methods as the focus here is to study the autocorrelation decays (in fact, reducing the stepsize leaves the autocorrelation decays indistinguishable). 100 different runs were averaged to reduce the sampling errors after the system was well equilibrated. The solid black line is the reference decay obtained by using Hamiltonian dynamics (*i.e.*, switching off the thermostat, $\gamma = 0$), which used exactly the same initial conformations and velocities as alternative stochastic dynamics.



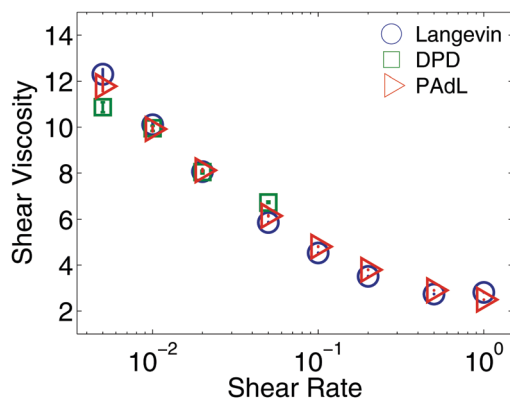


Fig. 4 Comparisons of the computed shear viscosity in a standard setting of polymer melts as described in Section 4.1 against shear rate by using various methods (in the low friction regime of $\dot{\gamma} = 0.5$) with Lees–Edwards boundary conditions. The simulation details of the figure are the same as in Fig. 1 except the same stepsize of $h = 0.01$ was used for all methods. Note that the error bars were included but could be seen with relatively small shear rates only. Note also that we did short runs to highlight the deviations, while the errors decrease further and the viscosity reaches the Newtonian plateau at small shear rates upon increasing the length of the runs, as is well known from previous studies.

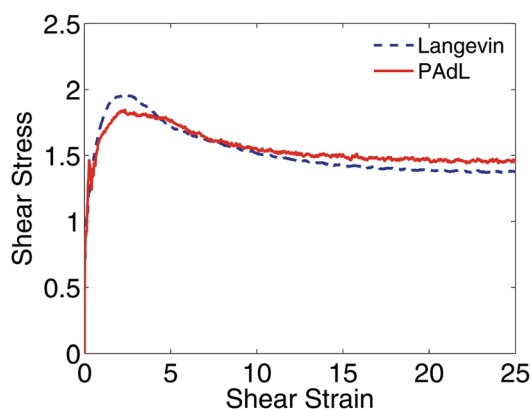


Fig. 5 Plot of the relations of shear stress and shear strain, which is the product of shear rate and simulation time, between Langevin dynamics and PAdL (in the low friction regime of $\dot{\gamma} = 0.5$) for the polymer melts modeled using Lees–Edwards boundary conditions with a shear rate of $\dot{\gamma} = 0.5$. The same stepsize of $h = 0.01$ was used for both methods. 10 000 different runs were averaged to obtain relatively smooth curves.

In Fig. 5, we plot the evolution of the shear stress against the shear strain for PAdL and Langevin. While PAdL was perfectly stable for all runs performed, Langevin dynamics occasionally becomes unstable with a shear rate of $\dot{\gamma} = 1$, therefore, a smaller shear rate of $\dot{\gamma} = 0.5$ was used here instead in order to provide comparisons. As can be seen from the figure, for both methods, as the shear strain increases, the shear stress rises rapidly from zero to its maximum at shear strain of around 2.5 before relaxing to its steady state, consistent with the results in Fig. 4. At startup the system tends to transform affinely and builds up stress before relaxation takes over. The faster the shear rate, the more likely the shear stress maximum occurs, as affine shear deformation. During this phase, particles are forced

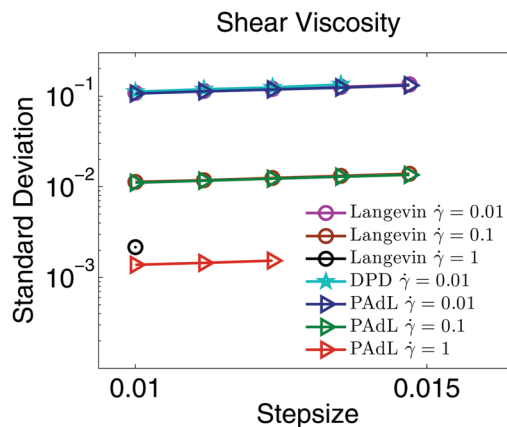


Fig. 6 Double logarithmic plot of the standard deviation in the computed shear viscosity at the specified rates against stepsize by using various methods (in the low friction regime of $\dot{\gamma} = 0.5$) in the presence of Lees–Edwards boundary conditions. The simulation details of the figure are the same as in Fig. 1 except the stepsizes tested began at $h = 0.01$.

into close proximity resulting in strong Lennard-Jones repulsion and quickly producing enormous shear stress. However, the maximum is not expected, and was indeed not observed by us, at small rates. Overall, the behavior is in good agreement with previous studies.^{89–92} We also investigated the evolution (including DPD whenever possible) with a wide range of shear rates and did not observe significant differences between the methods.

Fig. 6 plots the standard deviation in the computed shear viscosity over a wide range of shear rates by using various methods subject to Lees–Edwards boundary conditions. Once again, all methods behave very similarly. As we increase the shear rate, the standard deviation decreases, which is consistent with the observations in Fig. 4. Note that the DPD method

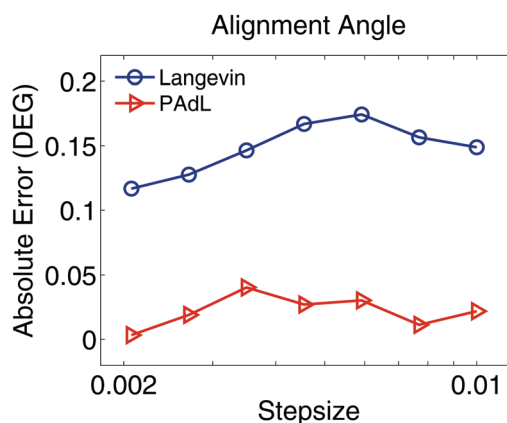


Fig. 7 For the polymer melts modeled using Lees–Edwards boundary conditions, the absolute error (in degrees) of the flow alignment angle (23) produced by the PAdL and Langevin algorithms (in the low friction regime of $\dot{\gamma} = 0.5$) is plotted against the stepsize (in semi-log scale) for a shear rate of $\dot{\gamma} = 1$. The reference value of the alignment angle was obtained by using the PAdL method with a sufficiently small stepsize of $h = 0.001$ (20 different runs). The simulation details of the figure are the same as in Fig. 1 except 20 different runs were averaged to further reduce the sampling errors.



appears only in the smallest shear rate case. In the regimes of small and moderate shear rates, the standard deviations of all/both methods are very similar. However, in the high shear rate ($\dot{\gamma} = 1$) case, Langevin dynamics has not only a visibly ($\approx 60\%$) larger standard deviation but also a smaller range of stepsizes usable than the PAdL method.

We further investigate in Fig. 7 the stepsize effects on the alignment angle of the polymer chains subject to LEBC at a relatively high shear rate of $\dot{\gamma} = 1$. It appears that while the absolute error of the alignment angle of PAdL remains below around 0.04 degrees, the corresponding error of Langevin dynamics is around six times larger. This clearly demonstrates the superiority of PAdL over Langevin dynamics in nonequilibrium simulations especially at relatively high shear rates.

5 Conclusions

We have reviewed a variety of numerical methods (SVV and BAOAB of Langevin dynamics, DPD, and PAdL) that can be used to simulate polymeric systems. We have systematically compared those methods in terms of accuracy, efficiency, and stability both in equilibrium and nonequilibrium settings.

In terms of sampling accuracy in equilibrium simulations, we have observed that the BAOAB and PAdL methods outperform the SVV and DPD methods in a wide range of friction coefficients. We have also discovered that while perfectly matching the reference decay, the OAF, which characterizes the orientational relaxation of the polymer chains, of both pairwise (momentum-conserving) thermostats (DPD and PAdL) has little dependence on the friction coefficient in a wide range. On the other hand, the OAF of Langevin dynamics strongly depends on the friction coefficient, moreover, a clear discrepancy was observed even for the commonly used low friction of $\gamma = 0.5$. We have further developed a careful procedure to quantify the sampling efficiency of various methods. By comparing the effective sample size, we found that PAdL substantially outperforms alternatives, particularly with a relatively small thermal mass of $\mu = 0.1$, for which remarkably about twenty times increase in the effective sample size was achieved in comparison to alternative approaches (see Table 1).

We are more focused on investigating the stability issues in nonequilibrium simulations. We have demonstrated that, with a stepsize of $h = 0.01$, the largest usable shear rate was around $\dot{\gamma} = 0.08$ for DPD, compared with $\dot{\gamma} = 1.2$ for both Langevin and PAdL, in a standard setting of polymer melts as described in Section 4.1. Thus, in agreement with previous studies,⁶¹ DPD is not recommended for nonequilibrium simulations, when the mean flow dissipation rates begin to overwhelm the thermostat, limiting its use in practice for relatively large shear rates. Between Langevin dynamics and PAdL, we have found that they perform rather similarly with relatively low shear rates. For the investigation of even smaller rates, where the FENE polymer exhibits a Newtonian plateau in the shear viscosity, thermodynamically guided methods are more suitable.^{93,94} Nevertheless, we have illustrated that while both methods share a similar relation of

shear stress and shear strain, Langevin dynamics performed unreliably with a relatively high shear rate of $\dot{\gamma} = 1$ at a stepsize of at least $h = 0.01$ —it not only produced a larger (by about 60%) standard deviation of the computed shear viscosity than PAdL, but also resulted in a significantly larger (about six times) absolute error of the flow alignment angle.

Conflicts of interest

There are no conflicts to declare.

Appendix

A Integration schemes

We list here detailed integration steps for both DPD-S1 and PAdL methods described in the article. Verlet neighbor lists⁷¹ are used throughout each method in order to reduce the computational cost.

Shardlow's splitting method: DPD-S1. For each interacting pair of particles i and j , the summations go over all interacting pairs within cutoff radius ($r_{ij} < r_c$),

$$\mathbf{p}_i^{n+1/4} = \mathbf{p}_i^n - \sum_{j>i} H_{ij} (\mathbf{e}_{ij}^n \cdot \mathbf{v}_{ij}^n) \mathbf{e}_{ij}^n + \sum_{j>i} \mathbf{J}_{ij},$$

$$\mathbf{p}_j^{n+1/4} = \mathbf{p}_j^n + \sum_{j>i} H_{ij} (\mathbf{e}_{ij}^n \cdot \mathbf{v}_{ij}^n) \mathbf{e}_{ij}^n - \sum_{j>i} \mathbf{J}_{ij},$$

$$\mathbf{p}_i^{n+2/4} = \mathbf{p}_i^{n+1/4} + \sum_{j>i} \mathbf{J}_{ij} - \sum_{j>i} \frac{H_{ij}}{1 + 2H_{ij}} \left[(\mathbf{e}_{ij}^n \cdot \mathbf{v}_{ij}^{n+1/4}) \mathbf{e}_{ij}^n + 2\mathbf{J}_{ij} \right],$$

$$\mathbf{p}_j^{n+2/4} = \mathbf{p}_j^{n+1/4} - \sum_{j>i} \mathbf{J}_{ij} + \sum_{j>i} \frac{H_{ij}}{1 + 2H_{ij}} \left[(\mathbf{e}_{ij}^n \cdot \mathbf{v}_{ij}^{n+1/4}) \mathbf{e}_{ij}^n + 2\mathbf{J}_{ij} \right],$$

where $H_{ij} \equiv \gamma \omega^D(r_{ij}^n) h/2$ and $\mathbf{J}_{ij} \equiv \sigma \omega^R(r_{ij}^n) \mathbf{e}_{ij}^n \sqrt{h} \mathbf{R}_{ij}^n / 2$ with \mathbf{R}_{ij}^n being normally distributed variables with zero mean and unit variance.

For each particle i ,

$$\mathbf{p}_i^{n+3/4} = \mathbf{p}_i^{n+2/4} + h \mathbf{F}_i^C(\mathbf{q}^n)/2,$$

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n + h \mathbf{v}_i^{n+3/4},$$

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^{n+3/4} + h \mathbf{F}_i^C(\mathbf{q}^{n+1})/2,$$

where $\mathbf{F}_i^C(\mathbf{q}) = -\nabla_{\mathbf{q}_i} U(\mathbf{q})$ are the total conservative forces acting on particle i with configuration \mathbf{q} .

Pairwise adaptive Langevin thermostat: PAdL. For each particle i ,

$$\mathbf{q}_i^{n+1/2} = \mathbf{q}_i^n + h \mathbf{v}_i^n/2,$$

$$\mathbf{p}_i^{n+1/4} = \mathbf{p}_i^n + h \mathbf{F}_i^C(\mathbf{q}^{n+1/2})/2.$$

The summations go over all interacting pairs within cutoff radius ($r_{ij} < r_c$),

$$\mathbf{p}_i^{n+2/4} = \mathbf{p}_i^{n+1/4} + \frac{1}{2} \sum_{j>i} m \Delta v_{ij} \left(\mathbf{q}^{n+1/2}, \mathbf{p}^{n+1/4}, \zeta^n \right) \mathbf{e}_{ij}^{n+1/2},$$

$$\mathbf{p}_j^{n+2/4} = \mathbf{p}_j^{n+1/4} - \frac{1}{2} \sum_{j>i} m \Delta v_{ij} \left(\mathbf{q}^{n+1/2}, \mathbf{p}^{n+1/4}, \zeta^n \right) \mathbf{e}_{ij}^{n+1/2},$$



with

$$\Delta v_{ij} = \begin{cases} (\mathbf{e}_{ij} \cdot \mathbf{v}_{ij}) [e^{-\tilde{\tau}h/2} - 1] + \sigma \sqrt{[1 - e^{-\tilde{\tau}h}]/(\xi m)} \mathbf{R}_{ij}, & \xi \neq 0; \\ (2\sigma/m)\omega^R(r_{ij})\sqrt{h/2}\mathbf{R}_{ij}, & \xi = 0, \end{cases}$$

where $\tilde{\tau} = 2\xi\omega^D(r_{ij})/m$ and \mathbf{R}_{ij} are normally distributed variables with zero mean and unit variance. For the additional variable ξ ,

$$\xi^{n+1} = \xi^n + hG(\mathbf{q}^{n+1/2}, \mathbf{p}^{n+2/4})/2,$$

where

$$G(\mathbf{q}, \mathbf{p}) = \mu^{-1} \sum_i \sum_{j>i} \omega^D(r_{ij}) [(\mathbf{v}_{ij} \cdot \mathbf{e}_{ij})^2 - 2k_B T/m].$$

The following summations go over all interacting pairs within cutoff radius ($r_{ij} < r_c$),

$$\mathbf{p}_i^{n+3/4} = \mathbf{p}_i^{n+2/4} + \frac{1}{2} \sum_{j>i} m \Delta v_{ij} (\mathbf{q}^{n+1/2}, \mathbf{p}^{n+2/4}, \xi^{n+1}) \mathbf{e}_{ij}^{n+1/2},$$

$$\mathbf{p}_j^{n+3/4} = \mathbf{p}_j^{n+2/4} - \frac{1}{2} \sum_{j>i} m \Delta v_{ij} (\mathbf{q}^{n+1/2}, \mathbf{p}^{n+2/4}, \xi^{n+1}) \mathbf{e}_{ij}^{n+1/2}.$$

For each particle i ,

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^{n+3/4} + h\mathbf{F}_i^C(\mathbf{q}^{n+1/2})/2,$$

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^{n+1/2} + h\mathbf{v}_i^{n+1}/2.$$

An extension of the PAdL algorithm for systems with beads of different masses and identical friction coefficients can be found in ref. 46.

Acknowledgements

The authors thank Michael Allen, Gabriel Stoltz, Karl Travis, and anonymous referees for valuable suggestions and comments. XS and BL acknowledge the support of the Engineering and Physical Sciences Research Council (UK) through Grant No. EP/P006175/1. Part of this work was done during XS and BL's stay at the Institut Henri Poincaré – Centre Émile Borel during the trimester “Stochastic Dynamics Out of Equilibrium”. XS and BL thank this institution for hospitality and support.

References

- 1 S. Plimpton, *J. Comput. Phys.*, 1995, **117**, 1–19.
- 2 D. Vlachakis, E. Bencurova, N. Papangelopoulos and S. Kossida, *Adv. Protein Chem. Struct. Biol.*, 2014, **94**, 269–313.
- 3 A. A. Hassanali, J. Cuny, V. Verdolino and M. Parrinello, *Philos. Trans. R. Soc., A*, 2014, **372**, 20120482.
- 4 R. H. Gee, N. Lavecic and L. E. Fried, *Nat. Mater.*, 2006, **5**, 39.
- 5 T. Kreer, *Soft Matter*, 2016, **12**, 3479–3501.
- 6 L. F. A. Bernardo, A. P. B. M. Amaro, D. G. Pinto and S. M. R. Lopes, *Comput. Mater. Sci.*, 2016, **118**, 32–46.
- 7 C. Li and A. Strachan, *J. Polym. Sci., Part B: Polym. Phys.*, 2015, **53**, 103–122.
- 8 A. Gooneie, S. Schuschnigg and C. Holzer, *Polymers*, 2017, **9**, 16.
- 9 Y. Li, B. C. Abberton, M. Kröger and W. K. Liu, *Polymers*, 2013, **5**, 751–832.
- 10 K. Z. Takahashi, R. Nishimura, K. Yasuoka and Y. Masubuchi, *Polymers*, 2017, **9**, 24.
- 11 R. H. Boyd, R. H. Gee, J. Han and Y. Jin, *J. Chem. Phys.*, 1994, **101**, 788–797.
- 12 V. A. Harmandaris, K. C. Daoulas and V. G. Mavrantzas, *Macromolecules*, 2005, **38**, 5796–5809.
- 13 T. A. Kavassalis and P. R. Sundararajan, *Macromolecules*, 1993, **26**, 4144–4150.
- 14 P. S. Stephanou, C. Baig and V. G. Mavrantzas, *Soft Matter*, 2011, **7**, 380–395.
- 15 J. Yang, X. Tang, Z. Wang, T. Xu, F. Tian, Y. Ji and L. Li, *J. Chem. Phys.*, 2017, **146**, 014901.
- 16 I.-C. Yeh, J. L. Lenhart, G. C. Rutledge and J. W. Andzelm, *Macromolecules*, 2017, **50**, 1700–1712.
- 17 A. Ramirez-Hernández, M. Müller and J. J. de Pablo, *Soft Matter*, 2013, **9**, 2030–2036.
- 18 K. Binder, T. Kreer and A. Milchev, *Soft Matter*, 2011, **7**, 7159–7172.
- 19 Z. Posel, M. Svoboda, C. M. Colina and M. Lísal, *Soft Matter*, 2017, **13**, 1634–1645.
- 20 K. Speyer and C. Pastorino, *Soft Matter*, 2015, **11**, 5473–5484.
- 21 Y. Li, M. Kröger and W. K. Liu, *Soft Matter*, 2014, **10**, 1723–1737.
- 22 A. Karatrantos, N. Clarke and M. Kröger, *Polym. Rev.*, 2016, **56**, 385–428.
- 23 T. C. Le, B. D. Todd, P. J. Daivis and A. Uhlherr, *J. Chem. Phys.*, 2009, **130**, 074901.
- 24 E. Córdova-Mateo, O. Bertran, B. Zhang, D. Vlassopoulos, R. Pasquino, A. D. Schlüter, M. Kröger and C. Alemán, *Soft Matter*, 2014, **10**, 1032–1044.
- 25 O. Bertran, B. Zhang, A. D. Schlüter, M. Kröger and C. Alemán, *J. Phys. Chem. B*, 2013, **117**, 6007–6017.
- 26 E. Córdova-Mateo, O. Bertran, A. D. Schlüter, M. Kröger and C. Alemán, *Soft Matter*, 2015, **11**, 1116–1126.
- 27 S. Raafatnia, O. A. Hickey and C. Holm, *Phys. Rev. Lett.*, 2014, **113**, 238301.
- 28 H. Merlitz, C. Li, C. Wu and J.-U. Sommer, *Soft Matter*, 2015, **11**, 5688–5696.
- 29 S. Das, M. Banik, G. Chen, S. Sinha and R. Mukherjee, *Soft Matter*, 2015, **11**, 8550–8583.
- 30 R. J. Loncharich, B. R. Brooks and R. W. Pastor, *Biopolymers*, 1992, **32**, 523–535.
- 31 M. Feig, *J. Chem. Theory Comput.*, 2007, **3**, 1734–1748.
- 32 G. Bussi and M. Parrinello, *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.*, 2007, **75**, 056707.
- 33 B. Peters, *Reaction Rate Theory and Rare Events*, Elsevier, 2017.
- 34 S. Nosé, *J. Chem. Phys.*, 1984, **81**, 511.
- 35 W. G. Hoover, *Phys. Rev. A: At., Mol., Opt. Phys.*, 1985, **31**, 1695.
- 36 G. Bussi, D. Donadio and M. Parrinello, *J. Chem. Phys.*, 2007, **126**, 014101.
- 37 G. Bussi and M. Parrinello, *Comput. Phys. Commun.*, 2008, **179**, 26–29.



- 38 G. Bussi, T. Zykova-Timan and M. Parrinello, *J. Chem. Phys.*, 2009, **130**, 074101.
- 39 B. Leimkuhler, E. Noorizadeh and O. Penrose, *J. Stat. Phys.*, 2011, **143**, 921–942.
- 40 M. P. Allen and F. Schmid, *Mol. Simul.*, 2007, **33**, 21–26.
- 41 R. G. Winkler, S. P. Singh, C.-C. Huang, D. A. Fedosov, K. Mussawisade, A. Chatterji, M. Ripoll and G. Gompper, *Eur. Phys. J.-Spec. Top.*, 2013, **222**, 2773–2786.
- 42 T. Maximova, R. Moffatt, B. Ma, R. Nussinov and A. Shehu, *PLoS Comput. Biol.*, 2016, **12**, e1004619.
- 43 E. Paquet and H. L. Viktor, *Biomed Res. Int.*, 2015, **2015**, 1–18.
- 44 A. Sokal, *NATO ASI Ser., Ser. B*, 1997, **361**, 131–192.
- 45 P. Hoogerbrugge and J. Koelman, *Europhys. Lett.*, 1992, **19**, 155.
- 46 B. Leimkuhler and X. Shang, *J. Comput. Phys.*, 2016, **324**, 174–193.
- 47 B. Dünweg, *J. Chem. Phys.*, 1993, **99**, 6977–6982.
- 48 V. Daggett, *Acc. Chem. Res.*, 2002, **35**, 422–429.
- 49 J.-L. Barrat, J. Baschnagel and A. Lyulin, *Soft Matter*, 2010, **6**, 3430–3446.
- 50 B. Leimkuhler and C. Matthews, *Appl. Math. Res. Express*, 2013, **2013**, 34–56.
- 51 B. Leimkuhler, C. Matthews and G. Stoltz, *IMA J. Numer. Anal.*, 2016, **36**, 13–79.
- 52 G. S. Grest and K. Kremer, *Phys. Rev. A: At., Mol., Opt. Phys.*, 1986, **33**, 3628.
- 53 K. Kremer and G. S. Grest, *J. Chem. Phys.*, 1990, **92**, 5057.
- 54 T. Soddemann, B. Dünweg and K. Kremer, *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.*, 2003, **68**, 046702.
- 55 S. Melchionna, *J. Chem. Phys.*, 2007, **127**, 044108.
- 56 B. Leimkuhler and C. Matthews, *J. Chem. Phys.*, 2013, **138**, 174102.
- 57 P. Español and P. Warren, *Europhys. Lett.*, 1995, **30**, 191.
- 58 R. D. Groot and P. B. Warren, *J. Chem. Phys.*, 1997, **107**, 4423.
- 59 C. Pastorino, T. Kreer, M. Müller and K. Binder, *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.*, 2007, **76**, 026706.
- 60 J. Cao and A. E. Likhtman, *Phys. Rev. Lett.*, 2012, **108**, 028302.
- 61 D. A. Fedosov, G. E. Karniadakis and B. Caswell, *J. Chem. Phys.*, 2010, **132**, 144103.
- 62 B. Leimkuhler and X. Shang, *J. Comput. Phys.*, 2015, **280**, 72–95.
- 63 T. Shardlow, *SIAM J. Sci. Comput.*, 2003, **24**, 1267–1282.
- 64 A. Brünger, C. L. Brooks III and M. Karplus, *Chem. Phys. Lett.*, 1984, **105**, 495–500.
- 65 D. J. Evans and G. P. Morriss, *Phys. Rev. Lett.*, 1986, **56**, 2172.
- 66 D. J. Evans and G. Morriss, *Statistical Mechanics of Non-equilibrium Liquids*, Cambridge University Press, 2008.
- 67 M. Whittle and K. P. Travis, *J. Chem. Phys.*, 2010, **132**, 124906.
- 68 A. Jones and B. Leimkuhler, *J. Chem. Phys.*, 2011, **135**, 084125.
- 69 B. Leimkuhler and X. Shang, *SIAM J. Sci. Comput.*, 2016, **38**, A712–A736.
- 70 X. Shang, Z. Zhu, B. Leimkuhler and A. J. Storkey, *Advances in Neural Information Processing Systems*, 2015, vol. 28, pp. 37–45.
- 71 L. Verlet, *Phys. Rev.*, 1967, **159**, 98.
- 72 B. A. Berg, *Markov Chain Monte Carlo Simulations and Their Statistical Analysis*, World Scientific, 2004.
- 73 J. Goodman and A. D. Sokal, *Phys. Rev. D: Part. Fields*, 1989, **40**, 2035–2071.
- 74 C. Aust, S. Hess and M. Kröger, *Macromolecules*, 2002, **35**, 8621–8630.
- 75 H. H. Rugh, *Phys. Rev. Lett.*, 1997, **78**, 772.
- 76 B. Butler, G. Ayton, O. G. Jepps and D. J. Evans, *J. Chem. Phys.*, 1998, **109**, 6519.
- 77 C. Braga and K. P. Travis, *J. Chem. Phys.*, 2005, **123**, 134101.
- 78 M. P. Allen, *J. Phys. Chem. B*, 2006, **110**, 3823–3830.
- 79 K. P. Travis and C. Braga, *J. Chem. Phys.*, 2008, **128**, 014111.
- 80 A. Lees and S. Edwards, *J. Phys. C: Solid State Phys.*, 1972, **5**, 1921.
- 81 J. Irving and J. G. Kirkwood, *J. Chem. Phys.*, 1950, **18**, 817–829.
- 82 M. S. Green, *J. Chem. Phys.*, 1954, **22**, 398–413.
- 83 R. Kubo, *J. Phys. Soc. Jpn.*, 1957, **12**, 570–586.
- 84 X. Shang, PhD thesis, University of Edinburgh, 2015.
- 85 M. Kröger, W. Loose and S. Hess, *J. Rheol.*, 1993, **37**, 1057–1079.
- 86 M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 1989.
- 87 J. D. Weeks, D. Chandler and H. C. Andersen, *J. Chem. Phys.*, 1971, **54**, 5237–5247.
- 88 M. Kröger, *Comput. Phys. Commun.*, 1999, **118**, 278–298.
- 89 J. Cao and A. E. Likhtman, *ACS Macro Lett.*, 2015, **4**, 1376–1381.
- 90 R. S. Hoy and M. O. Robbins, *Phys. Rev. Lett.*, 2007, **99**, 117801.
- 91 M. H. Wagner and J. Meissner, *Macromol. Chem. Phys.*, 1980, **181**, 1533–1550.
- 92 R. B. Bird, R. C. Armstrong and O. Hassager, *Dynamics of Polymeric Liquids. Volume 1: Fluid Mechanics*, Wiley-Interscience, 1987.
- 93 P. Ilg, H. C. Öttinger and M. Kröger, *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.*, 2009, **79**, 011802.
- 94 P. Ilg and M. Kröger, *J. Rheol.*, 2011, **55**, 69–93.

