No-slip boundary conditions and forced flow in multiparticle collision dynamics

Dan S. Bolintineanu,[*] Jeremy B. Lechman,[†] Steven J. Plimpton,[‡] and Gary S. Grest[§]
Sandia National Laboratories, Albuquerque, New Mexico 87185, USA

(Received 25 April 2012; published 12 December 2012)

Multiparticle collision dynamics (MPCD) is a particle-based fluid simulation technique that is becoming increasingly popular for mesoscale fluid modeling. However, some confusion and conflicting results persist in literature regarding several important methodological details, in particular the enforcement of the no-slip condition and thermostatting in forced flow. These issues persist in simple flows past stationary boundaries, which we exclusively focus on here. We discuss the parametrization of MPCD fluids and its consequences for fluid-solid boundaries in great detail, and show that the method of virtual particles proposed by Lamura *et al.* and adopted by many others is required only for parameter choices that lead to viscosities dominated by collisional contributions. We test several implementations of the virtual particle method and discuss how to completely eliminate slip at stationary boundaries. We also show that stochastic boundary reflection rules are inherently problematic for forced flow and suggest a possible remedy. Finally, we discuss the most robust way to achieve forced flow and evaluate several thermostatting methods in the process. All discussion is limited to solid objects that do not move as a result of collisions with MPCD particles (i.e., walls). However, the results can be extended to solutes that experience forces and torques due to interactions with MPCD particles (e.g., colloids). The detailed analysis presented for this simple case provides the level of rigor and accuracy to the MPCD method required for the study of more complex systems.

DOI: 10.1103/PhysRevE.86.066703 PACS number(s): 47.11.Mn, 83.10.Rs, 83.50.Rp

I. INTRODUCTION

Mesoscale fluid modeling techniques have received increased attention in recent years due to their relevance to many emerging nanotechnologies. In particular, the modeling of complex fluids at small length scales, where thermal fluctuations play an important role, has emerged as a problem of significant academic and technological interest. Numerous modeling techniques have been introduced in the last several decades to address this challenge, including dissipative particle dynamics (DPD), Lattice-Boltzmann simulations (LB), direct simulation Monte Carlo (DSMC), and various combinations of Brownian and Stokesian dynamics. In this work we focus on a simulation method closely related to DSMC, which is commonly referred to as stochastic rotation dynamics (SRD) or, in its more general form, as multiparticle collision dynamics (MPCD). All MPCD algorithms are based on modeling a fluid as a collection of point mass particles with no direct pairwise interactions. Alternate streaming and collision steps are used to simulate hydrodynamic behavior; because collisions conserve mass and momentum, the resulting fluid can be shown to obey the compressible Navier-Stokes equations for a Newtonian fluid [1,2]. A more detailed description of the key steps is deferred to the Methods and Background sections below. For an excellent survey of MPCD algorithms in general, including the underlying theoretical framework and numerous applications, the interested reader is referred to Refs. [3,4].

While the basic MPCD method is relatively well established, some confusion persists about several important algorithmic details, especially pertaining to the incorporation of solid boundaries. An enduring difficulty has been the enforcement of no-slip boundary conditions, such that the fluid velocity tangential to a solid object approaches zero near the solid boundary. This problem is not unique to MPCD but arises with most particle-based fluid dynamics simulation methods (e.g., DPD [5–8], LB [9,10], or SPH [11,12]). We focus exclusively on no-slip boundary conditions, since they are the most common choice for most engineering applications. For an in-depth analysis of partial slip in MPCD simulations, the interested reader is referred to recent work by Whitmer and Luijten [13].

This work aims to resolve some of the outstanding difficulties and confusion in enforcing no-slip boundary conditions in MPCD simulations and obtaining the correct fluid-solid interactions. We have found that a clear separation exists between the study of the flow field near solid boundaries and the study of the correct forces and torques on suspended solid objects. From an application perspective, the former case is relevant only in the case of boundaries that do not move in response to collisions with MPCD particles (i.e., walls), whereas the latter case is relevant in the case of objects that do (i.e., mixtures of MPCD particles and large solutes). We focus here on boundaries that do not experience forces or torques as a result of interactions with MPCD particles. In this way, we are able to isolate key details of fluid-boundary interactions in detail, as well as tackle a variety of issues pertaining to forced flow. The lessons learned from this simplified case can be extended to the case of suspended solutes. Following others, we simulate plane-Poiseuille flow between infinite parallel plates. Although we do not focus on applications here, we note that MPCD simulations of Poiseuille flow in simple geometries have recently been applied to the study of suspensions of red blood cells under flow [14] and polymer dynamics in microchannels [15–17]. For the present purposes, we use Poiseuille flow because a well-known analytical solution for the flow profile exists in this case, which allows us to readily

[*] dsbolin@sandia.gov
[†] jblechm@sandia.gov
[‡] sjplimp@sandia.gov
[§] gsgrest@sandia.gov

verify the accuracy of our simulation methods. Although the differences that arise between flow profiles with various settings are relatively small, the fact that a case as simple as Poiseuille flow has not been fully clarified raises some concerns. Small differences in such a simple system would likely result in significant errors in more complex systems, which is why we set out to resolve these issues with a high level of rigor and detail.

The paper is organized as follows: We first discuss the basic aspects of the MPCD algorithms relevant to the present work (Sec. II). We focus on the existence of different time step solutions for a particular parametrization, which has significant implications for flow fields near boundaries. In Sec. III A we introduce two methods of achieving forced flow, and in Sec III B we evaluate several thermostats that have been proposed for the stochastic rotation dynamics (SRD) collision algorithm. In Sec. III C we investigate two different methods that are commonly used to enforce no-slip boundary conditions, stochastic and reverse boundary reflections, and show the former to be problematic in forced flows. Section III D describes the effects of bin shifting, used to restore Galilean invariance. In Sec. III E we address the apparent wall slip that inevitably occurs when the fluid viscosity is dominated by collisional contributions and show results for several methods proposed to remedy this (virtual particles). We conclude in Sec. IV with a summary of the key results and offer a general prescription for MPCD simulations of forced flows.

II. METHODS AND BACKGROUND

A. Basic algorithms

The MPCD scheme consists of two key steps: in the first step, particles move at constant velocity, and positions are updated accordingly:

$$x_i^{t+1} = x_i^t + v_i \Delta t. \tag{1}$$

In the second step, particles are assigned to cubic bins that form an evenly spaced grid, and interparticle collisions are simulated by various momentum swapping schemes within each bin. These simulated collisions are not to be confused with collisions between MPCD particles and solid objects (walls or solutes). In the original SRD algorithm [1], interparticle collisions are simulated by rotating the velocities of the particles relative to the bin center-of-mass velocity. The velocities are thus updated according to

$$\mathbf{v}_{i,\xi}^{t+1} = \mathbf{u}_\xi^t + \mathbf{R}\big(\mathbf{v}_{i,\xi}^t - \mathbf{u}_\xi\big). \tag{2}$$

Here $\mathbf{R}$ is a stochastic rotation matrix and $\mathbf{u}_\xi$ is the center of mass velocity of all particles in bin ξ:

$$\mathbf{u}_\xi = \sum_{i=1}^{N_\xi} \frac{\mathbf{v}_{i,\xi}^t}{N_\xi}. \tag{3}$$

In two dimensions, the rotation is carried out by a fixed angle about the normal of the simulation plane, and the direction (clockwise or counterclockwise) is randomly selected. In three dimensions, which is the case that we are exclusively concerned with here, several rotation schemes have been proposed [18,19]. For computational expediency, we carry out rotations about one of six randomly selected orthogonal axes (aligned with the positive or negative *x, y,* and *z* axes), using a fixed rotation angle of 90 degrees. This is equivalent to "model B" in the work of Tüzel *et al.* [19]. A popular alternative is to select a randomly oriented rotation axis instead of picking only orthogonal axes. While there are some quantitative differences between the two approaches, all the conclusions that follow are qualitatively true for both methods.

Because the scheme in Eq. (2) conserves mass, linear momentum, and kinetic energy on a local, binwise basis, it can be shown that the SRD algorithm yields the correct hydrodynamics [1,2]. Furthermore, owing to the relative simplicity of the SRD algorithm, analytical expressions for the transport coefficients have been derived [19–22]. For our present purposes, we are only concerned with the shear viscosity of three-dimensional MPCD fluids, which consists of a kinetic contribution and a collisional contribution. For the SRD algorithm based on the rotation scheme described above, the viscosity is given by [19,23]

$$\begin{aligned}\nu_{\mathrm{SRD}} &= \nu_{\mathrm{SRD}}^{\mathrm{coll}} + \nu_{\mathrm{SRD}}^{\mathrm{kin}} \\ &= \frac{\Delta x^2}{18\Delta t}\left(1 - \frac{1-e^{-M}}{M}\right) + \frac{k_B T M \Delta t (M+2)}{4\rho\Delta x^3 (M-1)}.\end{aligned} \tag{4}$$

Here M is the mean number of SRD particles per bin, Δt is the time between collisions, Δx is the bin size, and ρ is the mass density of the fluid. Values of ν and ρ are chosen based on physical properties of the fluid, while values of Δx and M are selected based on numerical considerations. For a given set of these four values, Eq. (4) yields up to two distinct solutions for the time step Δt. This point is discussed in greater detail below.

Several additional collision rules have been proposed as alternatives to Eq. (2). Of particular interest are the multiparticle collision-Anderson thermostat (MPC-AT) and its angular-momentum conserving counterpart, referred to as MPC-AT+a [24,25]. Since we are not interested in rotational aspects of the flow in the present work, we defer any discussion of the MPC-AT+a algorithm to future work dealing with rotating solutes; the interested reader is referred to works by Götze and coworkers [26–28] and Downton and Stark [29].

In the MPC-AT algorithm, the particle velocities are updated according to

$$\mathbf{v}_{i,\xi}^{t+1} = \mathbf{u}_\xi + \mathbf{v}_{i,\mathrm{rand}} - \sum_{i=1}^{N_\xi} \mathbf{v}_{i,\mathrm{rand}}/N_\xi. \tag{5}$$

Here $\mathbf{v}_{i,\mathrm{rand}}$ represent velocities randomly drawn out of a Maxwell-Boltzmann distribution for each particle, and $\mathbf{u}_\xi$ is the mean velocity for a particular bin, defined as in Eq. (3). It can easily be shown that this scheme also conserves linear momentum, but it does not conserve kinetic energy. Instead, the collision rule (5) acts as a thermostat, where the set temperature corresponds to the temperature of the Maxwell-Boltzmann distribution of $\mathbf{v}_{i,\mathrm{rand}}$. An analytical expression for the viscosity of an MPC-AT fluid has been derived by Noguchi and Gompper. As in the case of the SRD collision rule, it can be separated into collisional and kinetic contributions:

$$\begin{aligned}\nu_{\text{MPC-AT}} &= \nu^{\text{coll}}_{\text{MPC-AT}} + \nu^{\text{kin}}_{\text{MPC-AT}} \\ &= \frac{\Delta x^2(M-1+e^{-M})}{12M\Delta t} \\ &\quad + \frac{k_BTM\Delta t}{\rho\Delta x^3}\left(\frac{M}{M-1+e^{-M}} - 0.5\right). \qquad (6)\end{aligned}$$

Early works that focused on proof-of-concept and method development [2,20,30,31] primarily used the original SRD algorithm suggested by Malevanets and Kapral [1]. More recently, the MPC-AT algorithm has become increasingly popular due to its simple extension to an angular momentum-conserving collision rule (MPC-AT+a) [25], which is required in cases where rotational flows and rotational solute motions are important. The principal drawback of the MPC-AT algorithm is that it tends to be more computationally expensive, approximately two times slower than the basic SRD algorithm for a pure fluid [3]. However, the MPC-AT algorithm implicitly includes a robust thermostat, whereas including a thermostat in the SRD algorithm is not trivial; depending on the choice of thermostat, this can result in a computational slowdown comparable to the MPC-AT algorithm [32]. Most simulations of Poiseuille flow have employed the SRD algorithm [1,31–33] with various thermostatting methods; however, Whitmer and Luijten [13] found that the MPC-AT algorithm, due to its inherently local thermostatting ability, yielded superior results when compared to the SRD collision rule coupled to a simple velocity rescaling scheme. Since both collision algorithms are viable options that have been used by various authors, we compare both in the present context.

B. Galilean invariance and mean free path limits

In both of the MPCD algorithms discussed so far, the presence of the grid of collision bins introduces an artificially fixed frame of reference, which breaks Galilean invariance. Furthermore, if the mean free path of particles, given by $\lambda = \Delta t\sqrt{k_BT/m_f} = \Delta t\sqrt{k_BTM/\rho\Delta x^3}$, is small relative to the size of the bins, particles within the same bin will undergo collisions with each other multiple times, leading to strong artifactual inter-particle correlations. In order to restore Galilean invariance and reduce correlations in the limit of a small mean free path [34], Ihle and Kroll [2,35] proposed shifting all bins by a random displacement between $-0.5\Delta x$ and $+0.5\Delta x$ prior to each collision (equivalently, particle positions can be randomly shifted within a fixed grid of bins, collisions can be carried out, and particles subsequently restored to their original positions). This bin-shifting procedure has several important implications for various aspects of MPCD simulations. Note that while bin shifting is not required in cases where the mean free path is greater than ~0.6× the bin size (i.e., $\lambda > 0.6\Delta x$), it can optionally be performed for any mean free path; in cases where the mean free path is small relative to the bin size, completely erroneous results are obtained if bin shifting is not used. Furthermore, bin shifting in the presence of solid walls will result in some bins being partially occupied by walls. These partial bins require special treatment, which we devote significant effort to in Sec. III E.

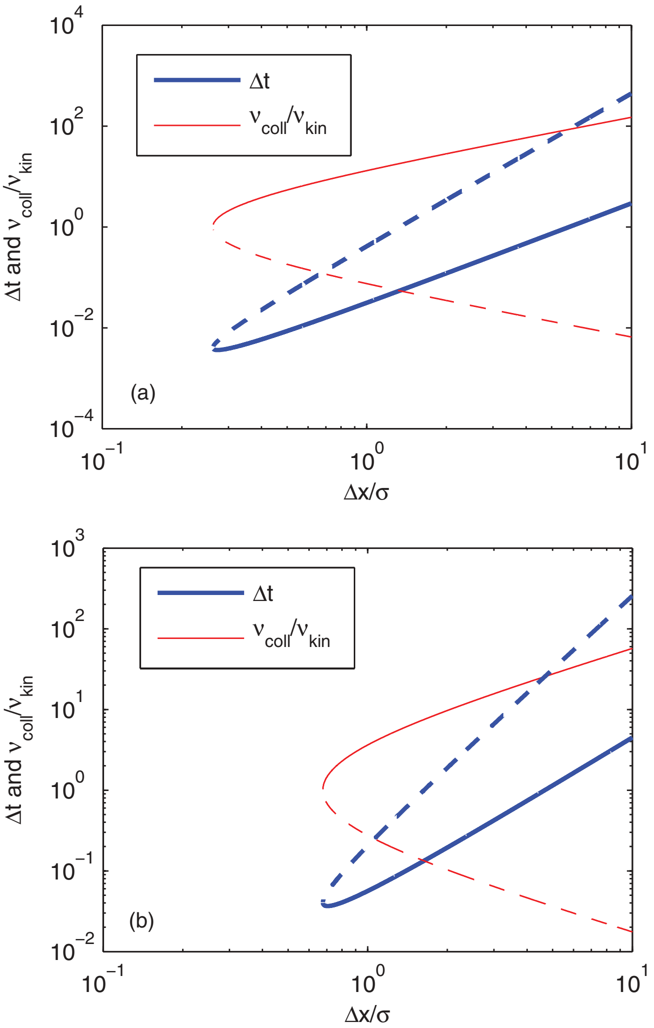

FIG. 1. (Color online) Time step Δt and $\nu_{\text{coll}}/\nu_{\text{kin}}$ as a function of bin size Δx for (a) the SRD collision algorithm and (b) the MPC-AT collision algorithm. The parameters correspond to those from Petersen *et al.* [39], which were selected to correspond to a Lennard-Jones fluid ($k_BT = 1\epsilon$, $\nu = 1.53\sigma^2/\tau$, $M = 5.28$, $\rho = 0.66m_{\text{LJ}}/\sigma^3$). The dashed lines indicate the branches corresponding to the large time step/large mean free path cases. Blue (thick) lines correspond to the Δt curves, red (thin) lines correspond to the $\nu_{\text{coll}}/\nu_{\text{kin}}$ curves.

As noted above, the viscosity expressions (4) and (6) allow up to two possible solutions for the time step Δt given a particular viscosity, density, and value of M. It should be emphasized, however, that these equations do not fully take into account the presence of correlations that may be present at small values of the mean free path; in this regime, the molecular chaos assumption breaks down, leading to deviations from the expected transport coefficients [36,37]. The effects of these correlations and suggestions for their correction have been investigated in some detail [21,34]. For the parameters used here (see Table I), we expect relatively small effects, even for the small mean free path case. As will be shown later, the viscosity, as measured both by the match to the Poiseuille flow profile and separately using the method of Müller-Plathe [38] (data not shown), is in excellent agreement with Eq. (4). Figure 1 shows the relationships between the

TABLE I. Summary of MPCD parameters used in this work.

	SRD		MPC-AT	
	Small λ	Large λ	Small λ	Large λ
$\Delta t = \lambda$	0.1220	3.476	0.1950	1.895
$\nu_{\rm coll}/\nu_{\rm kin}$	28.5	0.0351	9.72	0.103

bin size Δx and the time step Δt for a given viscosity of $\nu = 1.53\sigma^2/\tau$, $M = 5.28$ particles per bin, and a fluid density $\rho = 0.66 m_{\rm LJ}/\sigma^3$. Here σ represents the characteristic length scale, which in the case of a pure SRD fluid is reflected only in the bin size. This particular set of parameters was chosen by Petersen *et al.* [39] to approximate a Lennard-Jones fluid; the mapping procedure is described in detail in their paper [39]. Briefly, the values of the mass density ρ, viscosity ν, and temperature T were chosen to match those of a Lennard-Jones fluid. Numerically suitable values can then be used for M and Δx. A value of $M \sim 5$ is typical for computational efficiency in MPCD simulations; by selecting $M = 5.28$ and $\Delta x = 2.0\sigma$, the mass of MPCD particles can be made to match that of LJ particles ($m_f = m_{\rm LJ}$), while maintaining the desired density ($\rho = M m_f/\Delta x^3$). The time step solutions then follow from Eqs. (4) or (6). We use the same parameters here as they represent a realistic mapping to a fluid of interest and are also in the same range as typical MPCD parameters. In most previous SRD studies, convenient values were chosen for Δx, Δt, M, and m_f, resulting in comparable densities and viscosities, but without any correspondence to a particular fluid. Although the effects of the mean free path choice in MPCD have been studied extensively [19,20,36,37,40], the existence of two time step solutions for the *same* viscosity, density, bin size, and M has not been explicitly acknowledged. This is important in practical applications, where one typically wishes to simulate a fluid with a given viscosity and density. These fixed physical parameters, along with computational considerations for M and Δx, dictate the choice of the collision time step Δt, which typically has two possibilities, as discussed above.

The SRD and MPC-AT curves in Fig. 1 are plotted by solving Eqs. (4) and (6) for Δt given the parameters listed above. Since the temperature and particle mass are both set to unity here, the collision time steps are equal to the mean free paths. The ratios of the kinetic to collisional viscosities ($\nu_{\rm coll}/\nu_{\rm kin}$) are also shown in the same plot for both collision algorithms. We choose the same bin size as Petersen *et al.* [39], i.e., $\Delta x = 2\sigma$. For both SRD and MPC-AT, this leads to a small mean free path solution that corresponds to a total viscosity strongly dominated by the collisional component ($\nu_{\rm coll}/\nu_{\rm kin} = 28.5$ for SRD, 9.72 for MPC-AT). Conversely, the large mean free path solution yields a viscosity dominated by the kinetic component ($\nu_{\rm coll}/\nu_{\rm kin} = 0.0351$ for SRD, 0.103 for MPC-AT). It can easily be shown that this is always the case for both algorithms, and that $\nu_{\rm coll}/\nu_{\rm kin}$ is always equal to unity at the cusp of the Δt versus Δx curve.

We also plot the Schmidt numbers for the two algorithms in Fig. 2. The Schmidt number is defined as the ratio of momentum to mass transport, i.e., ${\rm Sc} = \nu/D$, where D is the self-diffusion coefficient of MPCD particles. Analytical expressions have been derived for D for both the SRD and MPC-AT collision rules [19,21,24]. The Schmidt number is often used as a robust measure of the dynamic characteristics of a fluid; low values of Sc correspond to a rarefied gas regime, whereas high values of Sc are associated with typical liquids. We note that in both cases, the large mean free path solution corresponds to a low Schmidt number [$O(1)$], whereas the small mean free path solution leads to larger Schmidt numbers ($\sim$10), which is close to the Schmidt number for a Lennard Jones fluid ($\sim$10) [41], but still lower than typical liquids ($\sim 10^2$–10^3). Depending on the system of interest, it may therefore be desirable to use the small time step solution for a particular choice of parameters to attain more liquid-like behavior. This topic has been discussed in great detail by Ripoll and coworkers [36,37] and to some extent by Padding and Louis [42]. The former study found that the persistence of hydrodynamic correlations in the higher Schmidt number cases (referred to by the authors as the "collective" regime) leads to deviations from the theoretically expected self-diffusion coefficient, presumably as a result of the breakdown of the molecular chaos approximation in this regime.

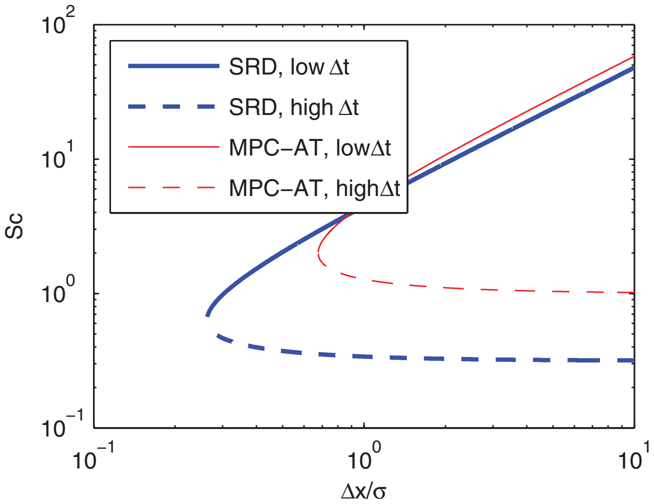

FIG. 2. (Color online) Schmidt number as a function of bin size Δx for the SRD and MPC algorithm ($\nu = 1.53\sigma^2/\tau$, $M = 5.28\sigma^{-3}$, $\rho = 0.66 m_{\rm LJ}/\sigma^3$). Blue (thick) lines correspond to curves for the SRD algorithm; red (thin) lines correspond to curves for the MPC-AT algorithm.

In the remainder of this work, we investigate several methodological details of MPCD simulations relating to flow near stationary boundaries: different methods of achieving forced flow, thermostatting methods for the SRD collision algorithm, the treatment of particle-wall collisions, and the effects of bins partially occupied by solids and their relationship to MPCD parameters. Each of these is addressed separately below, preceded by a brief review of the algorithmic details and previous work pertaining to each topic. We initially set out to investigate no-slip conditions due to their implications for moving particles, but it quickly became apparent that these are highly convoluted with the above-mentioned issues. Inconsistencies among methods used in the literature makes it difficult to isolate these effects, which motivated the exhaustive tests in the present work. All testing and development was carried out within the LAMMPS software [43]. The overall implementation of MPCD within LAMMPS has been described previously [39]. Novel features added in this work will be incorporated in the publicly available version of LAMMPS [44].

III. RESULTS AND DISCUSSION

A. Implementation of forced flow

In order to test the effects of various aspects of the SRD and MPC-AT algorithms, we conduct a series of simulations of Poiseuille flow between infinite parallel walls. A well-known analytical solution for the flow profile exists in this case, which serves as a clear test of the accuracy of the algorithm. Furthermore, numerous studies of this system have been previously reported in the literature as a means to evaluate various implementations and algorithmic details of MPCD fluids. This allows us to directly compare our results to existing ones. Assuming that the flow is in the x direction, and the gap between the walls is in the z direction, the velocity profile for Poiseuille flow is given by

$$v_x(z) = \frac{gz(L_z - z)}{2\nu}. \tag{7}$$

Here ν is the (total) kinematic viscosity, g is the forcing strength (e.g., acceleration due to gravity), and L_z is the separation between the walls in the z direction.

We have tested two distinct methods for forced flow. In the first method, which we refer to as direct forcing, the MPCD streaming step is modified to include an explicit constant acceleration on each particle. This also requires an update of the particle velocities, so that Eq. (1) is replaced with

$$r_i^{t+1} = r_i^t + v_i^t \Delta t + g\Delta t^2/2, \tag{8}$$

$$v_i^{t+1} = v_i^t + g\Delta t. \tag{9}$$

The strength of the forcing is controlled by g, the value of the acceleration.

In the second method, we designate an inlet region in which SRD particle velocities are assigned according to the analytical parabolic profile. We have tested several ways of assigning these velocities and have found that the best results are obtained if the assignment is carried out during the collision step, and the thermal component of each particle's velocity in the flow direction is retained. As long as the algorithm conserves kinetic energy, the flow profile will be sustained throughout the rest of the system, and no thermostatting is required. If we take the flow direction to be the x direction, the collision rule of Eq. (2) is modified to

$$\mathbf{v}_i^{t+1} = \begin{cases} g/(2\nu)z_i\left(L_z - z_i^t\right) + \mathbf{R}\left(\mathbf{v}_i^t - \mathbf{u}\right) & \text{if} \quad x_i < d_{\text{inlet}} \\ \mathbf{u} + \mathbf{R}\left(\mathbf{v}_i^t - \mathbf{u}\right) & \text{if} \quad x_i \geqslant d_{\text{inlet}} \end{cases}, \tag{10}$$

Here x_i^t and z_i^t are the x and z coordinates of particle i at the time of rotation, d_{inlet} is a cutoff value that defines the inlet region, and all other terms are the same as in Eq. (2). In subsequent discussions, the measured velocity profiles associated with this method correspond to averages taken beyond the inlet region (i.e., $x > d_{\text{inlet}}$). We refer to this scheme as the inlet flow method.

Early work by Malevanets and Kapral [1] presented two-dimensional Poiseuille flow results to demonstrate the general viability of the SRD algorithm. They enforced the desired velocity profile using the inlet method, which resulted in the expected parabolic velocity profile with no slip at the walls. Because their simulation was in a regime where the mean free path was large, no bin shifting was required, and the problem of partial bins was completely circumvented. In the work of Lamura *et al.* [31], two-dimensional Poiseuille flow was successfully simulated using the inlet flow method and the SRD algorithm. The first three-dimensional SRD forced flow simulations were carried out by Allahyarov and Gompper [18]. They report having tried both the inlet flow and direct forcing methods to achieve Poiseuille flow and found the former to be problematic. They therefore used the direct forcing method coupled with a simple velocity rescaling thermostat (whether the thermostat is applied globally or on a binwise basis is not clear). Subsequent work by Huang *et al.* [32], Whitmer and Luijten [13], and Imperio *et al.* [33] exclusively used the direct forcing method, coupled with various thermostatting schemes

TABLE II. Summary of MPCD simulations of Poiseuille flow from the literature.

Ref.	Flow/dim.	BCs/VPs	$\lambda/\Delta x$	Shift	Thermostat	$\frac{\nu_{\text{coll}}}{\nu_{\text{kin}}}$	Wall slip	Match to analytical expr.
[1]	Inlet/2D	Rev./None	0.11	Yes	None	430	Yes	Poor
[31]	Inlet/2D	Rev./None	0.11	Yes	None	430	Yes	Poor
	Inlet/2D	Rev./VP$_{\text{single}}$	0.11	Yes	None	430	None	~10% too low
	Inlet/2D	Rev./VP$_{\text{single}}$	0.63	No	None	13	None	Exact
[18]	Forced/3D	Rev./VP$_{\text{single}}$	0.046	Yes	Rescale	71	Minimal	Poor
	Forced/3D	Rev./VP$_{\text{single}}$	1.16	Yes	Rescale	0.11	Minimal	Poor
[32]	Forced/3D	Rev./VP$_{\text{single}}$	0.1	Yes	MBS	17	Yes	ν correct, slip
	Forced/3D	Rev./VP$_{\text{single}}$	0.1	Yes	Rescale (global)	17	Yes	ν 5% too low and slip
	Forced/3D	Rev./VP$_{\text{single}}$	0.1	Yes	None	17	Yes	ν 3% too high and slip
[13]	Forced/3D	Rev./VP$_{\text{single}}$	0.1	Yes	Rescale (global)	3.8	Yes	Close except slip
	Forced/3D	Rev./VP$_{\text{dens.}}$	0.1	Yes	Rescale (global)	3.8	Yes	Close except slip
	Forced/3D	Rev./VP$_{\text{dens.}}$	0.1	Yes	Rescale (local)	3.8	No	Exact
	Forced/3D	Rev./None	0.1	Yes	Rescale (global)	3.8	Yes	Close except slip
	Forced/3D	Stoch./VP$_{\text{dens.}}$	0.1	Yes	Rescale (local)	3.8	Yes	Close except slip
[33]	Forced/3D	Stoch./none	0.1	Yes	None	8.2	Large	Close except slip
	Forced/3D	Stoch./VP$_{\text{dens.}}$	0.1	Yes	None	8.2	Small	Close except slip

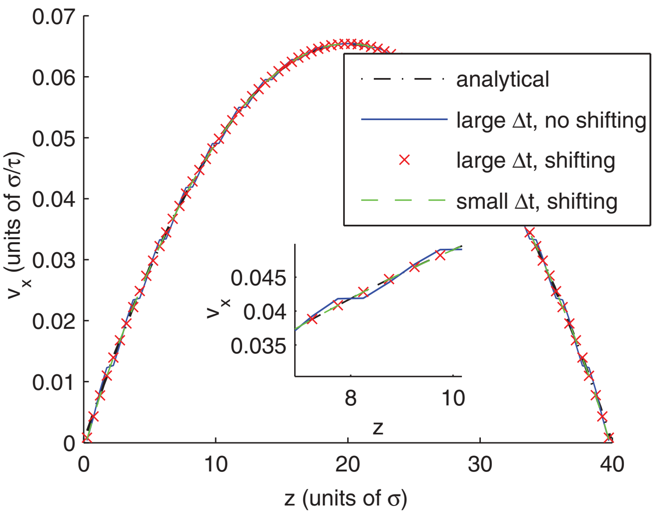

FIG. 3. (Color online) Velocity profiles for the inlet flow method ($\nu = 1.53\sigma^2/\tau$, $M = 5.28\sigma^{-3}$, $\rho = 0.66 m_{\rm LJ}/\sigma^3$, $\Delta x = 2.0$). Inset shows magnification of the velocity profile near the wall to illustrate jagged profile. All data are for the SRD collision algorithm with reverse BCs.

discussed below. These studies and the relevant parameters used therein are summarized in Table II.

Following Malevanets and Kapral [1] and Lamura *et al.* [31], we first carry out simulations of Poiseuille flow using the inlet flow method. We use the parameters discussed in Sec. II B. Results are shown for the SRD algorithm with reverse boundary conditions (BCs), where particle velocities are simply reversed upon colliding with the wall (a more detailed description of reverse BCs and alternative options are given in Sec. III C). For the inlet flow case, we used a forcing strength of $g = 0.0005\sigma/\tau^2$. The data correspond to a system size of $40\sigma \times 40\sigma \times 40\sigma$, for a total of 42,240 SRD particles. The inlet flow assignment is carried out in the region $x \leqslant 6\sigma$ [i.e., $d_{\rm inlet} = 6\sigma$ in Eq. (10)], and the velocity profile is only measured outside of this region. The velocity profile was allowed to equilibrate for 100 000 steps, after which measurements were taken. Sampling to construct the velocity profile in Fig. 3 was carried out every 10 steps, with histogram bins spaced 0.5σ apart (note that the sampling bins for the velocity profile are smaller than the rotation bins).

From the inset in Fig. 3, it can be seen that the velocity profile obtained with the large time step solution when shifting is not activated shows a steplike pattern, where the steps align with the SRD collision bins (see inset in Fig. 3; the steplike pattern is difficult to see even in the magnified profile but is more apparent in the high-resolution electronic version). This is likely a result of the manner in which the inlet flow is assigned [see Eq. (10)] and disappears when shifting is activated. Also, we note that there is no slip observed in any of the cases shown, even when the fluid viscosity is dominated by collisional contributions and partial bins arise due to shifting (see dashed green line in Fig. 3). Since the velocity profile is assigned at the inlet, rather than allowing the fluid to develop the velocity profile in response to a constant force, it appears that the effectively lower viscosity in partial bins does not result in slip. This is somewhat in disagreement with the results reported by Lamura *et al.* [31], who noted slip at the walls in similar conditions using the inlet flow method; however, their simulations were exclusively of two-dimensional systems, so that a direct comparison is difficult.

The inlet flow method fails for the MPC-AT collision algorithm (data not shown), as well as for an alternative particle-wall collision method that entails stochastic particle reflections (see Sec. III C). This is not surprising, since both introduce a stochastic element that one can expect violates local energy conservation and would cause the velocity profile assigned in the inlet region to become distorted downstream. Thus, it appears that the inlet flow method is not particularly robust, and we therefore turn to the direct forcing method for the remainder of this paper. Unless otherwise specified, we use the same parameters and system size as above for all the data that follow.

B. Forcing strength and thermostats

The direct forcing method coupled to the SRD algorithm requires the use of a thermostat to remove energy from the system due to the forcing term. No thermostat is required for MPC-AT, since the collision rule inherently maintains a constant temperature. We note that it is possible and physically justifiable to forgo the use of a thermostat in the case of forced flow coupled to the SRD algorithm: Thermalization can be enforced only at the walls, either through the use of stochastic boundary conditions or virtual particles, both of which will be discussed later. However, this approach will result in temperature and density variations across the channel. Since this will affect the resulting velocity profiles and we are only interested in isothermal flows, we use an explicit thermostat for all flow simulations based on the SRD collision algorithm. Several such thermostatting schemes have been proposed, which we briefly review here and evaluate in the context of forced flow.

All SRD thermostats involve a rescaling of the thermal component of velocity, which is equivalent to a modified collision rule, where Eq. (2) is replaced by

$$\mathbf{v}_i^{t+1} = u + s\mathbf{R}\big(\mathbf{v}_i^t - \mathbf{u}\big). \tag{11}$$

Here s is a scaling factor characteristic of each thermostatting method. In the simplest thermostat, which we refer to as simple velocity rescaling, the scaling factor s is determined from the ratio of the actual to the desired kinetic energies:

$$s = \sqrt{\frac{k_B T_{\rm set} d(n-1)}{m_f \sum_{i=1}^{n} \|\mathbf{v}_i - \mathbf{u}\|^2}}. \tag{12}$$

Here d is the number of dimensions that are being treated by the thermostat (which may be less than the dimensionality of the system; see below), n is the number of SRD particles in the bin, and $T_{\rm set}$ is the desired set point temperature. The local, binwise nature of this thermostat is essential in cases where well-defined flow profiles exist, so that only the thermal component of the velocity is affected, rather than rescaling based on a temperature that includes the flow velocity. We therefore do not consider the global version of this thermostat that has occasionally been used [13,42], where the sum in Eq. (12) is computed for the entire system, and the same scaling factor s is applied to all bins. However, even the binwise simple rescaling algorithm is a crude method that may still interfere with flow profiles locally, particularly if the bin length scale is not insignificant compared to variations in the flow profile, or if the number density of SRD particles is relatively low [24]. We note that in conventional molecular dynamics simulations, the use of an equivalent scheme as a global thermostat has been deprecated due to the problem of scaling based on bulk velocities rather than thermal velocities, leading to erroneous local thermal velocities (the "flying ice cube" problem [45]); however, in the case of SRD, where stochastic rotations ensure thorough momentum mixing and the thermostat is applied locally, this is not a concern. The rescaling thermostat has been used by several authors for forced flow SRD systems [13,18,42].

Two alternative, more sophisticated, SRD thermostats have been suggested. We explore these in some detail, since the thermostatting method has a strong effect on the accuracy of the flow profile. Hecht *et al.* [23] have suggested a Monte Carlo-like thermostat based on earlier work by Heyes [46]. The method is described in more detail in Refs. [3,23]. Briefly, a number ψ is selected between 1 and $1+c$ with uniform probability, where c is a small number between 0.05 and 0.3 that governs the strength of the thermostat. The scaling factor s is set to ψ or $1/\psi$ with a probability of 0.5 for each. The velocity rescaling in Eq. (11) is carried out according to a Metropolis-like criterion, with the probability of rescaling given by $p_A = \min(1, A)$, where

$$A = s^{d(n-1)} \exp[-m_f/(2k_B T_{\text{set}})] \sum_{i=1}^{n} \|\mathbf{v}_i - \mathbf{u}\|^2 (s^2 - 1). \tag{13}$$

This thermostat has been used successfully for the simulation of sedimenting colloids [23] and shown to yield rapid relaxation with minimal effects on fluid properties [3,23]. Additional applications of this thermostat include studies of the rheology [47], clustering [48], and microstructure morphologies [49] of colloidal suspensions under shear, in which the thermostat was used to remove energy added by imposed Couette flow.

We have also implemented the thermostat introduced recently by Huang *et al.* [32], and referred to as the Maxwell-Boltzmann scaling thermostat (MBS). In the MBS thermostat, an analytical expression is derived for the probability distribution of the kinetic energy of a particular bin:

$$P(E_k) = \frac{\exp(-E_k/k_B T)}{E_k \Gamma(d(n-1)/2)} [E_k/(k_B T)]^{3(n-1)/2}. \tag{14}$$

At each collision, a value of the kinetic energy E_k' is selected from the distribution (14) for each bin. The scaling factor for Eq. (11) is then set according to the ratio of this value to the measured kinetic energy of the bin:

$$s = \frac{2E_k'}{m_f \sqrt{\sum_{i=1}^{n} ||\mathbf{v}_i - \mathbf{u}^2||}}. \tag{15}$$

This algorithm has been shown to yield the correct velocity profile, temperature profile, and velocity distribution in Poiseuille flow conditions [32].

In molecular dynamics, it is well known that thermostats can interfere with flow profiles in an undesirable way. To minimize this interference, it has been suggested that thermostats only be applied to the nonflow directions, or only to the vorticity direction [50,51]. For all of the thermostats discussed above, it is possible to treat the flow, shear, and vorticity directions differently. We take the flow direction to be the x direction and the shear direction to be the z direction (i.e., the walls bound the system at $z = 0$ and $z = L_z$), which results in the vorticity direction being the y direction. For each SRD thermostat tested, it is possible to scale the components of the velocity (a) in all directions, (b) in the shear and vorticity directions only, and (c) in the vorticity direction only. Since all rescaling thermostats are ultimately empirical, none of these options can be excluded on clear theoretical grounds. We therefore test all three possibilities for all three thermostats.

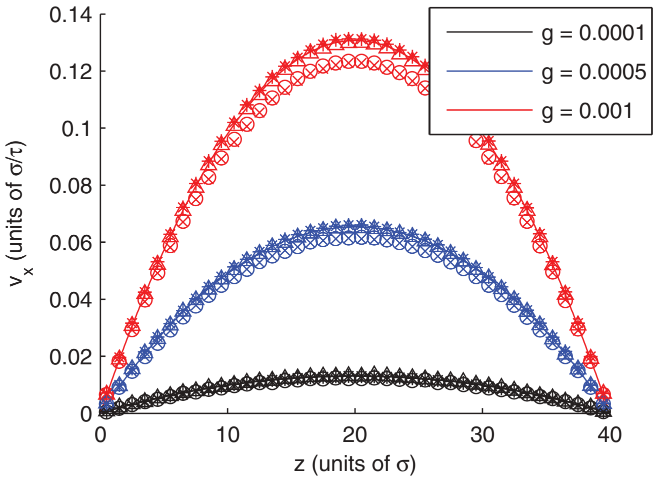

FIG. 4. (Color online) Comparison of various thermostats coupled with the SRD algorithm. Simple rescaling (×), Maxwell-Boltzmann scaling (o), Hecht *et al.* (△), MPC-AT (*). The lower set of curves corresponds to $g = 0.0001$, the middle set to $g = 0.0005$, and the higher set to $g = 0.001$.

Figure 4 shows the velocity profiles obtained using the SRD algorithm coupled with all three of the thermostats discussed (simple rescaling, Hecht *et al.* and Maxwell-Boltzmann scaling) three different forcing strengths ($g = 0.0001, 0.0005$, and $0.001\sigma/\tau^2$). In all cases, all three directions are included. For comparison, the velocity profiles generated using the MPC-AT algorithm are also plotted for the different forcing strengths, where the MPC-AT time step Δt is set to 1.895τ to match the viscosity of the SRD algorithm (recall that for SRD, $\Delta t = 3.476\tau$, $\nu = 1.53\sigma^2/\tau$). The data in Fig. 4 correspond to reverse BCs, where particle-wall collisions result in the reversal of the velocities of colliding SRD particles (see Sec. III C). The large mean free path limit is used for both algorithms, leading to viscosities dominated by kinetic contributions. Since the mean free path is sufficiently large ($\lambda > 0.6\Delta x$), no shifting is used in any of the simulations here. In all cases a system size of $40\sigma \times 40\sigma \times 40\sigma$ was used, and the equilibration period was 100 000 time steps followed by 500 000 production steps. Histogram sampling was carried out every 10 steps, with histogram bins spaced 0.5σ apart. We note that even at the high forcing strength, the maximum velocity at the center of the channel still corresponds to a low Mach number [$\text{Ma} = v_{\max}/v_{\text{sound}} = 0.131/(5/3\sqrt{k_B T/m}) = 0.08$]. As such, we do not expect compressibility effects to play any role here.

TABLE III. Comparison of different thermostats in terms of the percent deviation in the measured maximum velocity and the analytical result [see Eq. (16)].

	All			Shear and vorticity			Vorticity only		
Forcing	Low	Med.	High	Low	Med.	High	Low	Med.	High
Simple rescaling	−6.83	−6.02	−6.01	−0.58	−0.7	−1.07	1.97	0.51	0.39
MBS	−4.62	−5.63	−5.5	0.21	−0.44	−0.61	14.69	13.05	12.75
Hecht *et al.*	2.05	−0.58	−0.58	2.54	0.45	−0.05	1.73	0.663	−0.12
MPC-AT	0.13	1.02	0.43	N/A	N/A	N/A	N/A	N/A	N/A

While all the thermostats appear to perform well under low forcing conditions, the differences in their efficacy become increasingly apparent with increasing forcing strength. In particular, the simple rescaling and the MBS thermostat yield profiles that are too low, suggesting that they interfere with the flow too strongly. Note that in all cases, the analytical profiles were produced using Eq. (7) with a viscosity of $\nu = 1.53\sigma^2/\tau$.

The profiles shown in Fig. 4 correspond to the base case where all three spatial dimensions are coupled to the thermostat. As explained earlier, this may be problematic, as the thermostats in this case will directly influence the component of velocity in the flow direction, potentially resulting in a distortion of the velocity profile. Since the Hecht *et al.* thermostat is not applied at every collision step, it is no surprise that it does not suffer from the same shortcomings as the other two methods. Following this argument, we have also tested the other two thermostats in simulations where scaling is performed less frequently, but found that the resulting profiles are completely distorted even when the scaling is performed every third step (data not shown) rather than every step. Finally, we have tested all three thermostats in the cases where the velocity rescaling is performed on selective directions: as above, in all directions; only in the shear (z) and vorticity (y) directions; and only in the vorticity (y) direction. The velocity profiles are all parabolic and highly similar in most cases; for clarity, we forgo the full velocity profiles and instead tabulate the deviations from the analytical solutions in Table III, expressed as the fractional difference between the maximum velocity in the center of the channel as measured from the simulation and the analytically expected value:

$$\delta_v = \frac{v_{\text{max,meas.}} - v_{\text{max,analytical}}}{v_{\text{max,analytical}}} \times 100\%$$
$$= \frac{v_{\text{max,meas.}} - gL_z^2/8\nu}{gL_z^2/8\nu} \times 100\%. \tag{16}$$

Based on all these data, we identify the thermostat of Hecht *et al.* as the most robust with respect to reproducing the analytically expected velocity profile. Similar fidelity can be achieved for the simple rescaling thermostat if it is applied only in the vorticity direction, and for the MBS thermostat if applied in the vorticity and shear directions. In the case of the MBS thermostat applied only in the vorticity direction [$d = 1$ in Eq. (14)], drawing from the gamma distribution with $n = 2$ in Eq. (14) is problematic in our implementation, so thermostatting is not carried out in those cases. This may explain the poor performance of the MBS thermostat in the case where thermostatting is performed only in the vorticity direction. As expected, the MPC-AT algorithm, which contains an inherent thermostat in its collision scheme, performs well under all conditions.

C. Particle-wall collisions

The inclusion of solid objects in an MPCD fluid entails several difficulties. The first is associated with the streaming step, during which some MPCD particles will penetrate the surface of solid objects. These particle-wall collisions are not to be confused with the multiparticle collisions that occur in bins in the bulk fluid. In order to enforce no-slip boundary conditions at solid objects, two distinct methods are commonly used to assign the velocities of MPCD particles after collisions. The simplest and most intuitive of these is the reverse or bounce-back rule, in which all components of a colliding MPCD particle's velocity are reversed upon collision, i.e.,

$$\mathbf{v}_{\text{new}} = -\mathbf{v}_{\text{old}}. \tag{17}$$

More specifically, an MPCD particle that enters a solid object during a streaming step is first restored to the exact position and velocity at the point where it entered the object. In the case of collisions with stationary, flat walls, the exact amount of time that the particle traveled into the wall after the collision can easily be computed. The particle's velocity is then reversed [Eq. (17)], and the particle is streamed for the remainder of the time step with the new velocity. Note that in the case of forced flow, the acceleration of particles must be accounted for, both when restoring the particle to the collision point and when streaming the particle following velocity reversal. We refer to this scheme as reverse BCs throughout the remainder of the paper.

An alternative method of enforcing no-slip boundary conditions has been proposed by Inoue *et al.* [30]. Their method supposes that the solid object is filled with MPCD fluid particles at the same density as in the fluid. The outgoing particle velocity following a collision with a solid is then stochastically selected based on the distributions of the velocities of these imaginary particles as they exit the solid. The derivation of the underlying distributions was given in Ref. [30]; it is exact for a flat planar wall, and a good approximation when the mean free path of fluid particles is small relative to the radius of curvature of a curved solid object [23,30]. In practice, this means the velocity of an incoming particle is completely discarded, and the outgoing velocity is chosen to satisfy the following distributions:

$$p(v_n) \sim v_n \exp\left(-\beta v_n^2\right), \tag{18}$$
$$p(v_t) \sim \exp\left(-\beta v_t^2\right), \tag{19}$$

where v_n and v_t are the normal and tangential components of the velocity relative to the solid surface, $\beta = m_f/(2k_BT)$, and m_f is the mass of the MPCD fluid particles (taken here to be unity). As in the case of the reverse BCs above, the particle's position is returned to the exact point of contact with the wall and streamed with the new velocities for the remainder of the time step. In the case of forced flow, the acceleration must also be taken into account and applied in the final streaming step. We refer to this scheme as stochastic boundary conditions (BCs).

Both the stochastic and reverse method are in popular use. For simulations of colloids, it has been argued [23,42] that stochastic BCs represent more realistic collisions, since the surface of colloids is not typically smooth at the length scale of fluid particles. A fluid particle will likely undergo multiple, complex interactions with a colloid surface, resulting in an outgoing velocity that is completely uncorrelated to its incoming velocity. This behavior is approximately reproduced by stochastic BCs. These BCs have been studied in the context of colloid translational and rotational dynamics by Padding and coworkers [52]. In the context of forced flow, most authors have used reverse BCs [1,13,31,32] and observed some slip in cases where the viscosity was dominated by collisional contributions and partial bins were present. However, as we will show later, the effect in this case is not in fact a result of slip at the wall and is largely unrelated to the particle-wall reflection rule. Whitmer and Luijten [13] recently reported Poiseuille flow simulations with stochastic BCs and found that compared to reverse BCs with the exact same parameters, significant slip was observed in the stochastic BC case. Simulations by Imperio *et al.* [33] using stochastic BCs resulted in velocity profiles that match the expected parabolic profile, but with significant slip at the wall. If virtual particles (discussed below) were added, the slip was significantly diminished, but did not completely disappear. These studies and the relevant parameters used are summarized in Table II. We also note that in MPCD fluids, there are no issues with regard to layering of particles and finite size effects near walls; in other particle-based fluid simulation techniques (e.g., DPD [8]), the soft particle-particle and particle-wall interactions give rise to large density fluctuations near walls, which can lead to discrepancies between physical and hydrodynamic boundaries. Since MPCD particles are point particles that do not interact by any such potential with each other or with walls, these effects do not appear, and density profiles are completely uniform across the channel (data not shown).

All the results that have been presented so far correspond to the reverse BCs described by Eq. (17). We evaluate both reverse and stochastic BCs for their ability to reproduce the analytical Poiseuille flow profiles discussed above. The same simulation parameters are used, and the large mean free path limit is retained. For SRD, this implies a time step of $\Delta t = 3.476$, while for MPC-AT, the time step is $\Delta t = 1.895$. In both cases, the viscosity is dominated by kinetic contributions, and we therefore expect no notable effects of partial bins, as was the case in the thermostat tests. This allows us to isolate the effects of stochastic or reverse reflection rules independently of the influence of partial bins.

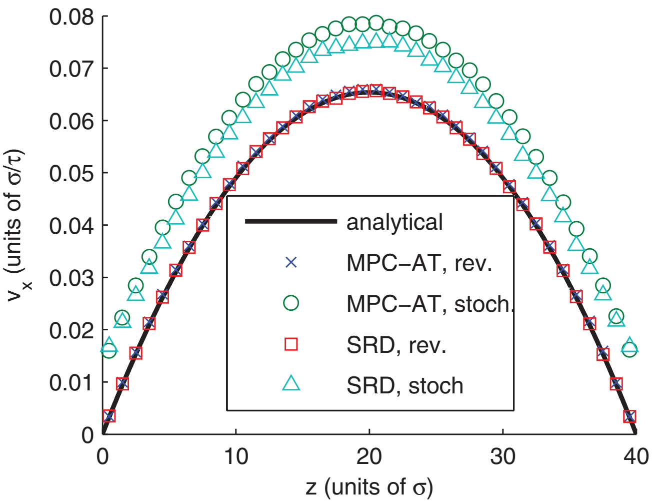

FIG. 5. (Color online) Stochastic and reverse BCs for the MPC-AT algorithm and the SRD collision algorithm coupled with the thermostat of Hecht *et al.* The usual parameter set was used ($\nu = 1.53\sigma^2/\tau$, $M = 5.28$, $\rho = 0.66m/\sigma^3$, $\Delta x = 2.0\sigma$) with the large mean free path solution ($\Delta t = 3.476\tau$ for SRD, 1.895τ for MPC-AT).

Figure 5 shows the velocity profiles obtained with the SRD algorithm coupled with the Hecht *et al.* thermostat using both reverse and stochastic BCs. A forcing strength of $g = 0.0005\sigma/\tau^2$ was selected, which yields a near-perfect match with the analytical profile in these conditions. Also shown are data for the MPC-AT fluid.

Clearly, significant slip occurs at the walls in the case of stochastic BCs, independently of any properties of the thermostat. We can understand this slip in terms of the average velocities of incoming and outgoing particles. In the case of reverse BCs, we expect the flow-direction component of the velocities of incoming particles to be equal and opposite in sign to those of outgoing (wall-reflected) particles by the very nature of reverse BCs. The average of the incoming and outgoing flow-direction velocities will then be zero, leading to the velocity measured at the wall being zero, thereby satisfying the no-slip condition. In the case of stochastic BCs, this is no longer the case, since particles arriving at the wall will have, on average, a significant nonzero velocity in the flow direction due to the forced flow. However, outgoing particles will have an average velocity of almost zero, since these velocities are drawn from a Maxwell-Boltzmann distribution with a mean of zero (the slight deviation from zero for outgoing particles is due to the acceleration they experience following the wall collision). Thus, in the case of stochastic BCs, the average of the incoming and outgoing particle velocities at the wall in the flow direction will be greater than zero, corresponding to the observed slip at the wall. In Figs. 6(a) and 6(b) we plot the incoming and outgoing flow-direction velocities for both types of boundary conditions along with their averages, as measured from the same simulations that produced the velocity profiles in Fig. 5. Plots are shown only for the SRD algorithm, but the same behavior is also observed in the case of MPC-AT (data not shown).

As expected, the average value for the reverse BCs is zero, whereas the stochastic BCs show a positive value (where positive velocities are taken to be in the flow direction, consistent with the velocity profile plots presented so far). Furthermore, the value of $0.5(v_{\rm in} + v_{\rm out})$ in the case of stochastic BCs is

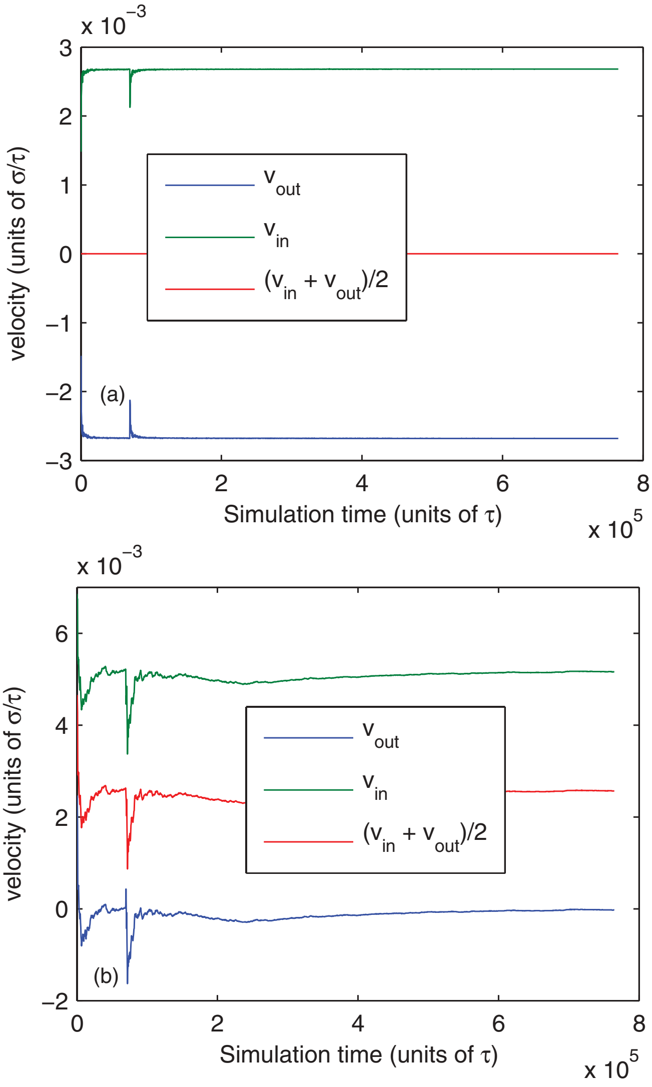

FIG. 6. (Color online) Particle velocities in the flow direction upon wall collision for (a) reverse BCs and (b) stochastic BCs. All data are running averages for the SRD collision algorithm with low forcing ($g = 0.0001\sigma/\tau^2$). In both cases, the top line correponds to v_{in}, the middle line to $(v_{\text{in}} + v_{\text{out}})/2$, and the bottom line to v_{out}.

equal to the magnitude of the slip observed in the measured velocity profile (compare to Fig. 5). It is important to note that this is a problem unique to forced flow, where a sustained flow profile exists outside of the solid boundary. In simulations of colloids in thermal equilibrium with no flow, colliding particles do not generally have a non-zero average incoming velocity in any direction, so that the effect disappears.

Our results suggest a possible remedy for the slip observed in forced flow with stochastic BCs. If the outgoing particle velocities are biased in the flow direction to be on average equal and opposite to the velocities of incoming particles, the no-slip condition can be achieved. Adding this bias is equivalent to shifting the mean of the Maxwell-Boltzmann distribution that the tangential components of velocities are drawn from [see Eq. (19)]. Its value can be determined directly as the average of the velocities of incoming particles in the desired, no-slip case. This value can then be taken as the average incoming particle velocity in the reverse BC case, where the correct profile is established. Figure 7 shows the velocity profiles corresponding to stochastic BCs corrected in this manner, showing that the slip does indeed disappear.

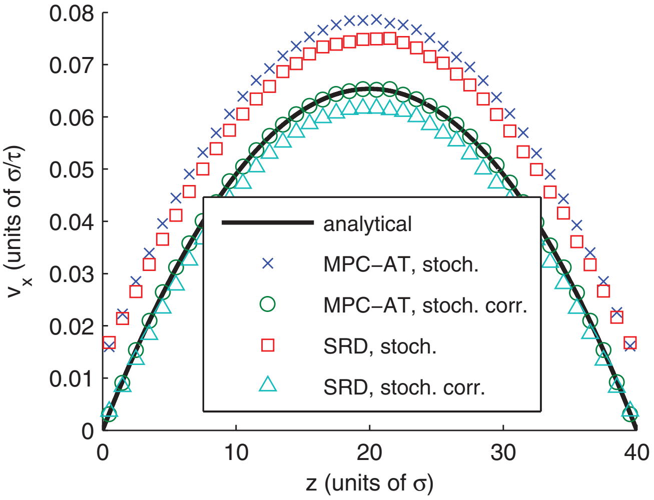

FIG. 7. (Color online) Corrected stochastic BCs for both collision algorithms.

D. Effects of bin shifting

As discussed in Sec. II B, both SRD and MPC methods can be employed with random bin shifting to restore Galilean invariance. In all cases discussed so far, the mean free path of the particles has been larger than the collision bin size Δx, so that bin shifting was not required (typically, $\lambda > 0.6\Delta x$ is sufficient to ensure that interparticle correlations decay sufficiently fast). However, we have found that the use of bin shifting in the cases discussed above results in a small difference in the velocity profile. Results are shown in Fig. 8 for the same parameter set as above, with bin shifting active.

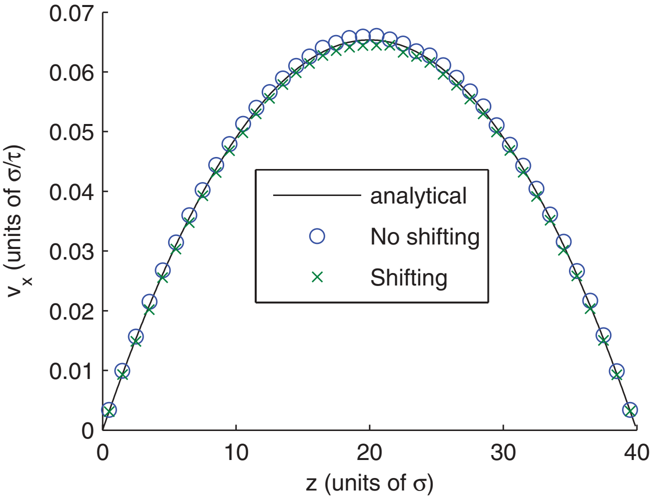

FIG. 8. (Color online) The effects of bin shifting on the velocity profile. Results correspond to the MPC-AT algorithm with a forcing strength of $g = 0.0005\sigma/\tau^2$. The usual parameters were used ($\nu = 1.53\sigma^2/\tau, M = 5.28, \Delta x = 2.0\sigma, \Delta t = 1.895\tau$).

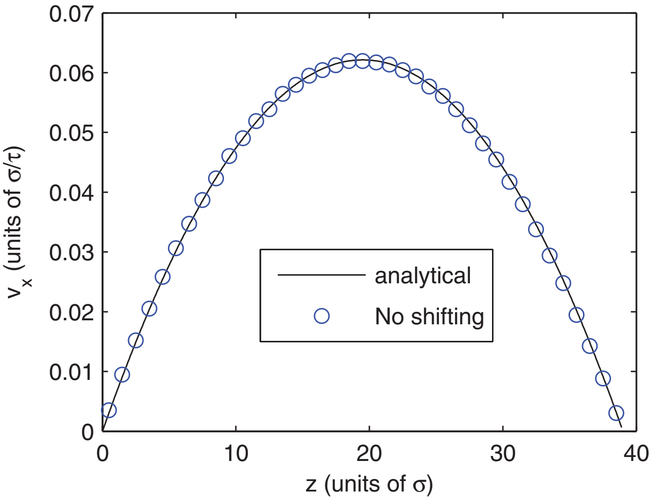

FIG. 9. (Color online) Velocity profile for the same parameters as in Fig. 8, with no shifting and partial bins resulting from positioning the top wall at $z = 39\sigma$.

We show data for the MPC-AT collision algorithm in the large mean free path limit ($\Delta t = 1.895\tau$) with a forcing strength of $g = 0.0005\sigma/\tau^2$. Similar results are obtained for the SRD algorithm and other forcing strengths (data not shown).

Clearly, bin shifting results in a slightly lower velocity profile, indicating a slightly higher viscosity. We exclude thermostatting as a probable cause for this, since the effect is present in the MPC-AT algorithm. Furthermore, we can exclude the effects of the partial bins by carrying out a similar simulation, where bin shifting is turned off, but partial bins are produced by setting one of the wall positions to coincide with the center of the collision bins.

In Fig. 9 we plot the velocity profile for a system with the same parameters as above and no shifting, except that the top wall is placed at $z = 39\sigma$ instead of $z = 40\sigma$. Since the bin dimension is 2σ, this means that partially filled bins will arise at the top wall. Since the match to the analytical profile is good in this case, with no discernible slip, we can safely ascertain that the effect in Fig. 8 is not a result of partial bins. More likely, shifting causes a slight increase in the shear viscosity, even in cases where the mean free path is large relative compared to the bin size. We have tested this hypothesis using the Müller-Plathe [38] method to measure viscosities. The method has been applied by Petersen *et al.* [39] to measure the viscosity of an SRD fluid; the interested reader is referred to their work for further details. From our data, it appears that bin shifting generally yields slightly higher viscosities (on the order of $<1\%$; data not shown), but the difference is not sufficient to explain the results in Fig. 8, nor is the accuracy of our Müller-Plathe viscosity measurements sufficient to draw any strong conclusions with such small differences. Since this effect is relatively minor, we do not pursue it further here. More importantly, we have shown in this section that in the case of a large mean free path and a viscosity dominated by kinetic contributions, there is no discernible slip as a result of partial bins, whether they arise due to bin shifting (Fig. 8) or are inherent in the system geometry (Fig. 9).

E. Virtual particles and partial bins

In the previous section, we addressed concerns associated with incorporating a solid boundary in MPCD simulations specific to the streaming step, during which particles can enter solid boundaries. The second aspect of fluid-solid interactions is more subtle and deals with the effects of binning on the ability of a fluid to resolve a solid boundary. In MPCD algorithms, binwise multiparticle collisions form the basis of the fluid coarse-graining procedure, and bins are roughly analogous to fluid elements in a discrete representation of the fluid. As such, one can expect that solid boundaries that do not align with bin boundaries will be problematic. This will inevitably occur with any suspended solid object that moves continuously through the discrete collision grid of MPCD bins, or with any irregularly shaped object. In the present case, this misalignment of solid boundaries and bin boundaries can be avoided if the flat walls are placed contiguously to bin boundaries; however, the introduction of bin shifting will inevitably disrupt this alignment and result in bins that are only partially filled with fluid.

The effect of partial bins was recognized relatively early in the development of the MPCD method, and the first suggestion for a correction came from work by Lamura *et al.* [31]. They suggested adding "virtual" particles to the solid region: At each collision, several additional MPCD particles are generated inside the solid region and assigned velocities randomly from a Maxwell-Boltzmann distribution. In their initial implementation, the number of virtual particles added was simply the difference between the actual number of particles in a partial bin and the mean number of particles in a bin in the bulk fluid, i.e.,

$$n_{\mathrm{VP}} = M - n_{\mathrm{MPCD}}. \tag{20}$$

Note that n_{VP} here may take on noninteger values, whereas a negative value simply means that no virtual particles are added. In the implementation of Lamura *et al.* instead of adding multiple particles, a single massive particle of mass $(M - n_{\mathrm{MPCD}})m_f$ is added, and no position is assigned to it [since none is required for the collision schemes (2) and (5)]. The coupling of the virtual particle to the real MPCD particles is achieved by modifying Eq. (3) so that the average velocity for bin ξ becomes

$$\mathbf{u}_\xi = \sum_{i=1}^{N_\xi} \frac{\mathbf{v}_{i,\xi}^t + \mathbf{a}}{M}. \tag{21}$$

Here $\mathbf{a}$ is the velocity of the virtual particle, whose components are drawn from a Gaussian distribution with variance $(M - n_{\mathrm{MPCD}})k_B T/m_f$ [equivalent to a Maxwell-Boltzmann distribution with mass $m_f/(M - n_{\mathrm{MPCD}})$]. In this and all other cases, virtual particles participate in the MPCD collisions as though they were regular particles. Lamura *et al.* [31] showed that this significantly ameliorates apparent slip in Poiseuille flow simulations and results in the correct flow field around a variety of solid boundaries. We refer to this implementation of virtual particles as $\mathrm{VP}_{\mathrm{single}}$, alluding to its use of a single massive particle.

Subsequent work introduced a number of similar schemes to correct the effects of partial bins. Several of these have been discussed by Whitmer and Luijten [13]. We implement and test two virtual particle schemes in addition to the VP$_{\text{single}}$ method discussed above. In the first of these, the number of virtual particles to be added to each partial bin is chosen such that the total number of particles in that bin (including both real MPCD particles and virtual particles) follows a Poisson distribution with mean M. Unlike the VP$_{\text{single}}$ method, this allows for multiple virtual particles within the solid and ensures that not only the mean of the number of particles in the partial bins matches that in the bulk fluid, but also its distribution. Virtual particles have the same mass as real MPCD particles, and they are included in the calculation of the mean bin velocity and collision rule exactly in the same way that real MPCD particles are. We refer to this method as VP$_{\text{Poisson}}$. The third virtual particle implementation that we include is similar to the VP$_{\text{Poisson}}$ method in its use of explicit virtual particles, but differs in the way these are distributed. Instead of binwise selection and positioning of virtual particles, the third method simply distributes virtual particles randomly throughout any solid region, while ensuring that the overall virtual particle number density matches the number density of the bulk fluid. For computational expediency, virtual particles in this implementation are only distributed within the solid walls to a depth of $\sqrt{3}\Delta x$. This depth is equal to the longest diagonal of the collision bins, and is the largest distance that a collision bin can extend into a general-shape solid object and still include the fluid region. We refer to this method as VP$_{\text{density}}$. Additional studies that employ virtual particles and the particular implementation used are summarized in Table II.

We have so far only dealt with the large mean free path solution discussed in Sec. II B, corresponding to viscosities dominated by kinetic contributions. It is therefore not surprising that partial bins, whether they arise due to bin shifting or the placement of walls, have not resulted in slip at the wall. The only slip that we have noted so far has been due to the use of stochastic BCs, and we have shown this to be an inherent feature of this reflection scheme that can be corrected with an appropriate modification. We now turn our attention to the small mean free path limit, which corresponds to the collisional-dominated viscosities for both the SRD and MPC-AT algorithm (refer to Table I and accompanying discussion).

Figure 10 shows the velocity profiles obtained using the SRD algorithm coupled to the Hecht thermostat, using a collision time step of $\Delta t = 0.122\tau$, as well as the equivalent profile for the MPC-AT thermostat, using a collision time step of $\Delta t = 0.195\tau$. As previously discussed, this yields a total viscosity of $\nu = 1.53\sigma^2/\tau$ for both algorithms, which is dominated by collisional contributions in both cases. All other parameters are kept at the same values as the simulations discussed above to facilitate a more direct comparison. A system size of $40\sigma \times 40\sigma \times 40\sigma$ was used in both cases, with a forcing strength of $g = 0.0005\sigma/\tau^2$, and the same equilibration strategy as before. Sampling was carried out with histogram bins of size 1.0σ and 0.1σ, with the latter only shown in the inset to clarify the velocity profile near the boundary. We use reverse BCs in all the data that follow in order to isolate the effects peculiar to partial bins from those associated with stochastic BCs. No virtual particles are used in generating the data in Fig. 10.

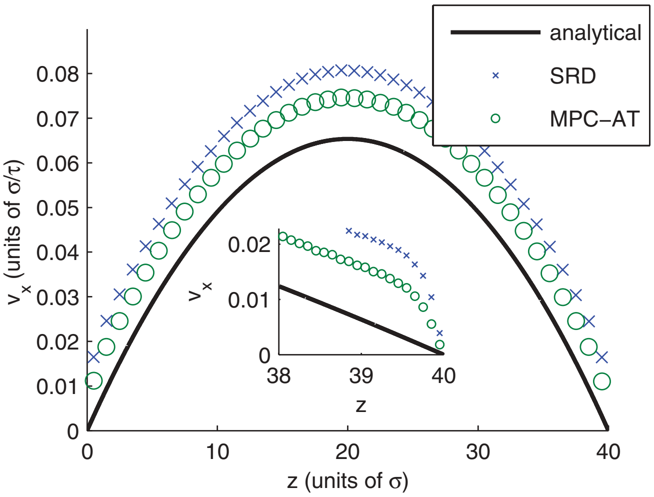

FIG. 10. (Color online) Velocity profiles for both algorithms in the case of the small mean free path. The usual parameters were used ($\nu = 1.53\sigma^2/\tau$, $M = 5.28$, $\rho = 0.66 m_{LJ}/\sigma^3$, $\Delta x = 2.0\sigma$), with the small time step solutions for both algorithms ($\Delta t = 0.122\tau$ for SRD, $\Delta t = 0.195\tau$ for MPC-AT). Bin shifting and reverse BCs are used in all cases.

Judging by the data in the main figure, corresponding to sampling with histogram bins of size 1.0σ, it appears that both the SRD and MPC-AT cases experience significant slip at the walls. However, a closer look reveals that this apparent slip is in fact a result of a much steeper velocity profile near the wall, suggesting a lower viscosity in these regions than the nominal bulk value of $1.53\sigma^2/\tau$. This is shown in the inset in Fig. 10, where the same velocity profiles are plotted with a much higher sampling resolution of 0.1σ. This is consistent with the explanation given by Whitmer and Luijten [13] for the slip observed in their simulations in the absence of virtual particles, where they argue that the occurrence of partially filled bins will result in lower effective collisional viscosities. Intuitively, the lower number of particles in bins adjacent to the wall can be expected to bring about a reduction in viscosity. Using a kinetic theory argument, Whitmer and Luijten showed that the viscosity in partial bins decreases to $1/3$ of its bulk value in the limit of large M, with even larger decreases for typical values of M.

This problem was originally recognized by Lamura *et al.* [31], who were the first authors to suggest including virtual particles to remedy the apparent slip in such cases. This has now become a relatively standard technique, with different authors adopting different methods of including virtual particles. In Fig. 11(a) we plot the velocity profiles obtained using the SRD collision algorithm coupled to the Hecht *et al.* thermostat [23], with all three virtual particle implementations discussed above. The same data are plotted for the MPC-AT algorithm in Fig. 11(b). As before, the inset in each figure corresponds to the velocity profiles near the wall, with a spatial sampling resolution of 0.1σ.

Clearly, the addition of virtual particles significantly ameliorates the apparent slip at the walls that occurs in Fig. 10. However, a very close look at the velocity profile near the wall (see insets in Fig. 11) reveals that the slip is not

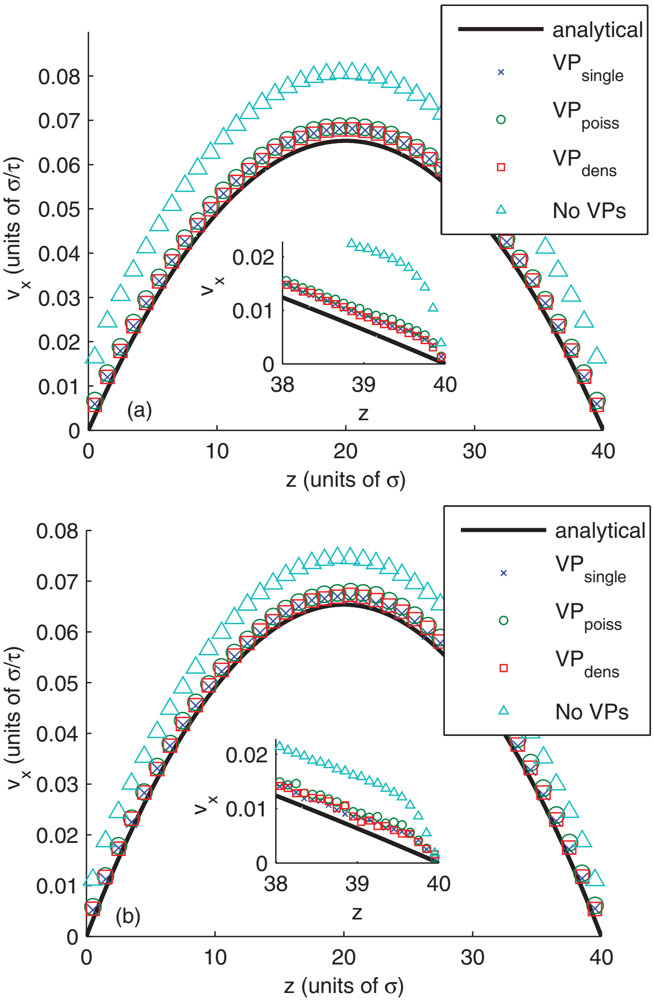

FIG. 11. (Color online) Velocity profiles for different virtual particle implementations for (a) the SRD algorithm, with $\Delta t = 0.122\tau$ and (b) the MPC-AT algorithm, with $\Delta t = 0.195\tau$. The time step solution corresponding to the small mean free path/collisional-dominated viscosity is used in both cases. All other parameters are the same as before ($\nu = 1.53\sigma^2\tau$, $\rho = 0.66 m_{LJ}/\sigma^3$, $M = 5.28$, $\Delta x = 2.0\sigma$). Bin shifting and reverse BCs were used.

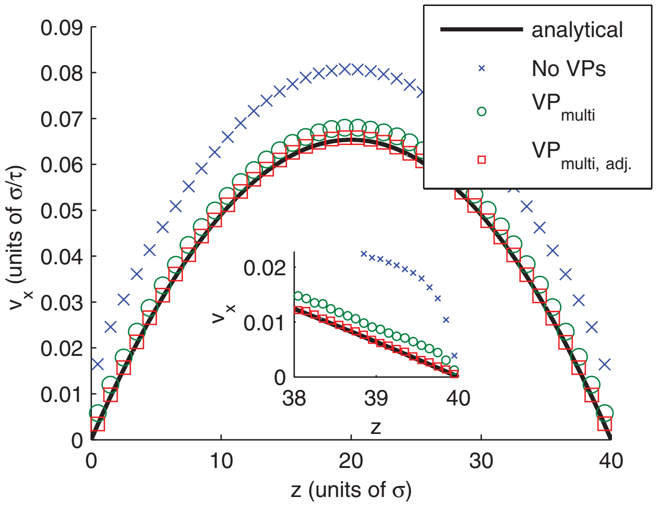

FIG. 12. (Color online) Velocity profiles obtained with virtual particles velocities adjusted based on the flow profile extrapolated into the wall. Only data for the SRD collision algorithm with the VP_{multi} method are shown, but results are similar for the MPC-AT algorithm and other virtual particle methods.

completely removed. This is the same result obtained by Imperio *et al.* [33] and Huang *et al.* [32], both of whom report simulations of Poiseuille flow in the small mean free path/collision-dominated viscosity regime and include virtual particles for partially occupied bins. In the work of Whitmer and Luijten [13] or the work of Allahyarov and Gompper [18] it is unclear whether virtual particles completely remove the slip, or whether some small slip persists, as we and others have found.

We posit that the remaining slip is caused by virtual particles not being coupled to the flow field, but instead having an average velocity of zero in the flow direction. This means that in the collision step, the mean bin velocity [see Eq. (3)] includes virtual particles that experience zero flow, resulting in a net positive velocity in the flow direction for that bin. As a result, the partially filled bins fail to approximate bulk fluid bins, where all particles are affected by the flow field and mean bin velocities reflect an average that includes the flow field throughout the bin. To address this, we propose assigning virtual particles velocities that have components in the flow direction based on the analytical Poiseuille flow profile [Eq. (7)]. This entails extrapolating the analytical solution of the Poiseuille flow profile into the solid wall, and using the velocity at a given virtual particle location as the mean of the distribution from which the x component of the virtual particle velocities are drawn. Equivalently, for each virtual particle, one simply evaluates the analytical Poiseuille flow formula [Eq. (7)] at the z coordinate of that virtual particle; the resulting velocity (which is negative) is then added to the x component of the velocity of that virtual particle.

In both the $\text{VP}_{\text{single}}$ and $\text{VP}_{\text{poisson}}$ methods, virtual particles are not assigned positions within the bin, since only their velocities contribute to the collision scheme. However, the VP_{multi} method requires that particles be assigned positions as well as velocities. As such, the modification that we propose extends most naturally to the VP_{multi} method, although it is certainly possible to extend the other two methods and assign positions to virtual particles.

In Fig. 12 we plot the resulting velocity profiles for the VP_{multi} method using the same parameters that produced the data in Fig. 11, with the addition of the Poiseuille flow analytical correction to the virtual particle velocities.

Clearly, the correction to the virtual particle velocities to account for the flow field eliminates the remaining slip (see inset in Fig. 12). Unfortunately, this correction requires a priori knowledge of the flow field near the solid boundary, which may render it impractical for some systems. However, just as in the case of the analogous problem with stochastic BCs, this problem is specific to forced flow, and will not have a significant effect on diffusing solutes, where no persistent flow fields exist near the solid surface.

IV. CONCLUSIONS

We have presented an in-depth analysis of several important details pertaining to the incorporation of solid boundaries in MPCD simulations, using both the MPC-AT and the SRD collision algorithms. The primary goal of this work has been to explain and remedy the occurrence of slip at solid boundaries in MPCD fluid simulations. We have focused exclusively on boundaries that do not move as a result of collisions with fluid particles, as this greatly simplifies the analysis and allows us to easily identify and measure slip. As part of our effort, we have also elucidated several other aspects of MPCD simulations of forced flow. Although differences among velocity profiles obtained with different methods are seemingly small, the underlying problems that we have identified are likely to be exacerbated in more complex systems.

All the tests presented herein are based on Poiseuille flow between parallel flat walls. The simplicity of this system has allowed us to isolate key algorithmic details that have only been cursorily addressed in the literature. We have enforced flow both by assigning the desired profile at the channel inlet and by directly applying a constant acceleration to all particles during streaming. Although the former approach has the advantage of not requiring a thermostat, it fails when stochastic elements that do not conserve kinetic energy locally are introduced, such as stochastic BCs or the MPC-AT collision algorithm. Furthermore, because velocities are prescribed at the inlet, the measured profile downstream will simply reflect the assigned profile, rather than capturing the response of the fluid and the underlying physical characteristics of the system. We therefore recommend the forced flow method in simulations of such systems, which is more robust and more closely corresponds to physical systems of interest.

Since the addition of a constant forcing term requires the use of a thermostat, we have evaluated several thermostatting methods that have been proposed for the SRD algorithm. While many criteria can be adopted to gauge the efficacy of various thermostats, we have focused on the ability of each thermostat to reproduce the correct Poiseuille flow profile, since this is the most relevant result to the present context. We have found that the thermostat suggested by Hecht *et al.* [23] generally performs best in this regard. We have shown that the simple binwise rescaling thermostat and the Maxwell-Boltzmann scaling thermostat suggested by Huang *et al.* [32] interfere with the flow field more strongly, but this effect can largely be ameliorated by excluding the flow and shear directions from thermostatting. Given the robustness of the Hecht *et al.* thermostat, we have used it in all subsequent forced flow SRD simulations. However, we cannot conclude that this is the optimal choice in all situations; since all SRD thermostats discussed are ultimately empirical, one should ideally evaluate all options for their ability to reproduce various properties of interest on a case by case basis.

Building on the analysis of Petersen *et al.* [39] we have shown that for both the SRD and MPC-AT algorithms, the choice of MPCD parameters will have a significant impact on the treatment of fluid-solid interactions. In particular, given a particular set of fluid properties (viscosity, density, and number of MPCD particles per bin), we have explored the relationship between the MPCD bin size and time step. For both algorithms, typical bin size choices of order unity lead to two time step solutions, corresponding to small and large mean free paths relative to the bin size. We have selected the bin size such that the small and large time step solutions correspond to viscosities dominated by collisional and kinetic contributions, respectively. We then showed that for the large mean free path solution, no-slip boundary conditions are readily achieved provided that reverse wall reflections are used, regardless of the choice of collision algorithm or the presence of partial bins. In the case of stochastic wall reflections, we have shown that slip at the wall is inherently present, but can be remedied by an appropriate adjustment of the outgoing particle velocities. However, this is only a concern where persistent flow fields exist outside of the solid boundary (as is the case in forced flow).

In the last section of this work, we have addressed the small mean free path cases for both algorithms, which correspond to viscosities dominated by collisional contributions. In agreement with others [13,31–33], we note significant slip for both collision algorithms even with reverse BCs. The addition of virtual particles significantly ameliorates this slip, but does not completely eliminate it. We have tested three common implementations of virtual particles and have found them to be almost equivalent for the present case; with moving boundaries, particularly where the angular momentum conserving version of the MPC-AT algorithm is used, we expect different virtual particle implementations to make a more significant difference. We have explained the small slip that persists even in the presence of virtual particles as being caused by the exclusion of virtual particles from the flow field. We have proposed a simple correction, wherein the velocities of the virtual particles are set based on the known local flow profile, and have shown that this does indeed eliminate the remaining slip. This requires knowledge of the local flow field and the ability to extrapolate it into the solid; hence, it may be limited in its applications. However, as with stochastic BCs, it should be problematic only in cases where persistent flow exists, such as the forced flow considered here.

The analysis presented has dealt exclusively with stationary boundaries, and no mention has been made of the forces applied on the solid surfaces by the fluid. In most applications of interest, this latter point is crucial, as it often dominates the behavior of suspended solutes. The primary purpose of this work was to lay the groundwork for such an investigation, and elucidate several aspects of MPCD simulations pertaining to solid boundaries. We plan to extend the lessons learned herein to an investigation of moving boundaries in the near future.

ACKNOWLEDGMENTS

This work was performed, in part, at the Center for Integrated Nanotechnologies, a US Department of Energy, Office of Basic Energy Sciences user facility. Funding for this work was provided through the National Institute for Nano-Engineering and by the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia National Laboratories is a multiprogram laboratory managed and operated by Sandia Corporation, a Lockheed-Martin Company, for the US Department of Energy's National Nuclear Security Administration under Contract No. DE-AC04-94AL85000.

[1] A. Malevanets and R. Kapral, J. Phys. Chem. **110**, 8605 (1999).
[2] T. Ihle and D. M. Kroll, Phys. Rev. E **67**, 066705 (2003).
[3] G. Gompper, T. Ihle, D. M. Kroll, and R. G. Winkler, Adv. Polym. Sci. **221**, 1 (2009).
[4] R. Kapral, in *Advances in Chemical Physics*, Vol. 140, edited by S. A. Rice (John Wiley & Sons, New York, 2008) pp. 89–146.
[5] S. M. Willemsen, H. C. J. Hoefsloot, and P. D. Iedema, Int. J. Mod. Phys. C. **11**, 881 (2000).
[6] M. Revenga, I. Zuniga, P. Espanol, and I. Pagonabarraga, Int. J. Mod. Phys. C. **9**, 1319 (1998).
[7] I. V. Pivkin and G. E. Karniadakis, J. Comput. Phys. **207**, 114 (2005).
[8] J. Smiatek, M. P. Allen, and F. Schmid, Eur. Phys. J. E **26**, 115 (2008).
[9] D. P. Ziegler, J. Stat. Phys. **71**, 1171 (1993).
[10] A. J. C. Ladd, J. Fluid. Mech. **271**, 311 (1994).
[11] F. Macia, M. Antuono, L. M. Gonzalez, and A. Colagrossi, Prog. Theor. Phys. **125**, 1091 (2011).
[12] D. W. Holmes J. R. Williams, and P. Tilke, Int. J. Numer. Anal. Methods Geomech. **35**, 419 (2011).
[13] J. K. Whitmer and E. Luijten, J. Phys.: Condens. Matter **22**, 104106 (2010).
[14] H. Noguchi and G. Gompper, Proc. Nat. Acad. Sci. USA **102**, 14159 (2005).
[15] L. Cannavacciuolo, R. G. Winkler, and G. Gompper, Europhys. Lett. **83**, 34007 (2008).
[16] R. Chelakkot, R. G. Winkler, and G. Gompper, Europhys. Lett. **91**, 14001 (2010).
[17] R. Chelakkot, R. G. Winkler, and G. Gompper, J. Phys.: Condens. Matter **23**, 184117 (2011).
[18] E. Allahyarov and G. Gompper, Phys. Rev. E **66**, 036702 (2002).
[19] E. Tuzel, M. Strauss, T. Ihle, and D. M. Kroll, Phys. Rev. E **68**, 036701 (2003).
[20] T. Ihle and D. M. Kroll, Phys. Rev. E **67**, 066706 (2003).
[21] T. Ihle, E. Tuzel, and D. M. Kroll, Phys. Rev. E **72**, 046707 (2005).
[22] N. Kikuchi, C. M. Pooley, J. F. Ryder, and J. M. Yeomans, J. Chem. Phys. **119**, 6388 (2003).
[23] M. Hecht, J. Harting, T. Ihle, and H. J. Herrmann, Phys. Rev. E **72**, 011408 (2005).
[24] H. Noguchi and G. Gompper, Phys. Rev. E **78**, 016706 (2008).
[25] H. Noguchi, N. Kikuchi, and G. Gompper, Europhys. Lett. **78**, 10005 (2007).
[26] I. O. Gotze, H. Noguchi, and G. Gompper, Phys. Rev. E **76**, 046705 (2007).
[27] I. O. Gotze and G. Gompper, Europhys. Lett. **92**, 64003 (2010).
[28] I. O. Gotze and G. Gompper, Phys. Rev. E **82**, 041921 (2010).
[29] M. T. Downton and H. Stark, J. Phys.: Condens. Matter **21**, 204101 (2009).
[30] Y. Inoue, Y. Chen, and H. Ohashi, J. Stat. Phys. **107**, 85 (2002).
[31] A. Lamura, G. Gompper, T. Ihle, and D. M. Kroll, Europhys. Lett. **56**, 319 (2001).
[32] C. C. Huang, A. Chatterji, G. Sutmann, G. Gompper, and R. G. Winkler, J. Comput. Phys. **229**, 168 (2010).
[33] A. Imperio, J. T. Padding, and W. Briels, Phys. Rev. E **83**, 046704 (2011).
[34] E. Tuzel, T. Ihle, and D. M. Kroll, Phys. Rev. E **74**, 056702 (2006).
[35] T. Ihle and D. M. Kroll, Phys. Rev. E **63**, 020201 (2001).
[36] M. Ripoll, K. Mussawisade, R. G. Winkler, and G. Gompper, Phys. Rev. E **72**, 016701 (2005).
[37] M. Ripoll, K. Mussawisade, R. G. Winkler, and G. Gompper, Europhys. Lett. **68**, 106 (2004).
[38] F. Muller-Plathe, Phys. Rev. E **59**, 4894 (1999).
[39] M. K. Petersen, J. B. Lechman, S. J. Plimpton, G. S. Grest, P. J. in 't Veld, and P. R. Schunk, J. Chem. Phys. **132**, 174106 (2010).
[40] T. Ihle, E. Tuzel, and D. M. Kroll, Phys. Rev. E **70**, 035701 (2004).
[41] K. Meier, A. Laesecke, and S. Kabelac, J. Chem. Phys. **121**, 9526 (2004).
[42] J. T. Padding and A. A. Louis, Phys. Rev. E **74**, 031402 (2006).
[43] S. Plimpton, J. Comp. Phys. **117**, 1 (1995).
[44] http://lammps.sandia.gov.
[45] S. C. Harvey, R. K. Z. Tan, and T. E. Cheatham, J. Comput. Chem. **19**, 726 (1998).
[46] D. M. Heyes, Chem. Phys. **82**, 285 (1983).
[47] M. Hecht, J. Harting, M. Bier, J. Reinshagen, and H. J. Herrmann, Phys. Rev. E **74**, 021403 (2006).
[48] M. Hecht, J. Harting, and H. J. Herrmann, Int. J. Mod. Phys. C. **18**, 501 (2007).
[49] M. Hecht, J. Harting, and H. J. Herrmann, Phys. Rev. E **75**, 051404 (2007).
[50] P. A. Thompson and M. O. Robbins, Phys. Rev. Lett. **63**, 766 (1989).
[51] P. A. Thompson and M. O. Robbins, Phys. Rev. A **41**, 6830 (1990).
[52] J. T. Padding, A. Wysocki, H. Lowen, and A. A. Louis, J. Phys.: Condens. Matter **17**, S3393 (2005).