

A unified operator splitting approach for multi-scale fluid-particle coupling in the lattice Boltzmann method

Ulf D. Schiller¹

*Theoretical Soft Matter and Biophysics, Institute of Complex Systems,
Forschungszentrum Jülich, 52425 Jülich, Germany*

Abstract

A unified framework to derive discrete time-marching schemes for coupling of immersed solid and elastic objects to the lattice Boltzmann method is presented. Based on operator splitting for the discrete Boltzmann equation, second-order time-accurate schemes for the immersed boundary method, viscous force coupling and external boundary force are derived. Furthermore, a modified formulation of the external boundary force is introduced that leads to a more accurate no-slip boundary condition. The derivation also reveals that the coupling methods can be cast into a unified form, and that the immersed boundary method can be interpreted as the limit of force coupling for vanishing particle mass. In practice, the ratio between fluid and particle mass determines the strength of the force transfer in the coupling. The integration schemes formally improve the accuracy of first-order algorithms that are commonly employed when coupling immersed objects to a lattice Boltzmann fluid. It is anticipated that they will also lead to superior long-time stability in simulations of complex fluids with multiple scales.

Keywords: lattice Boltzmann, fluid-particle coupling, force coupling, immersed boundary method, external boundary force

1. Introduction

The lattice Boltzmann method (LBM) [1] is a popular approach to simulate hydrodynamic phenomena and is widely used as an alternative to continuum-based approaches to solving the Navier-Stokes equations. The LBM is a mesoscopic model that bridges between kinetic theory and macroscopic hydrodynamics, i.e., it reproduces the conservation laws of the Boltzmann equation (in particular mass and momentum conservation) on macroscopic time and length

Email address: u.schiller@ucl.ac.uk (Ulf D. Schiller)

¹Present address: Centre for Computational Science, University College London, 20 Gordon Street, London WC1H 0AJ, United Kingdom

scales [2–5]. Since the LBM enables simple and accurate incorporation of geometric boundaries, it is widely applied to fluid-structure interaction problems [6–8].

In recent years, there has been growing interest in using the lattice Boltzmann method to simulate momentum transport and hydrodynamic interactions in complex fluids and soft matter systems [9, 10]. Such systems concern objects that are immersed in a continuum solvent, for example, colloidal suspensions [11, 12], polymer solutions [13–15], or biological cells [16–18]. These objects can be represented by a set of “coarse-grained” particles which interact via effective potentials and reproduce the correct dynamics on physically interesting scales. Since these scales are typically large compared to atomistic scales, the solvent can be treated as a continuum and the hydrodynamic fields represent slow degrees of freedom of the microscopic kinetics. The standard lattice Boltzmann algorithm gives a second-order accurate approximation of the macroscopic transport equations for the conserved moments, however, for grid-scale Reynolds numbers above unity the non-conserved moments may be subject to oscillations that can cause nonlinear instabilities [19]. Soft matter flows are usually in the creeping flow regime, i.e., they are incompressible (low Mach number) and inertia can be neglected (low Reynolds number).

The dynamics of complex fluids with immersed objects can be strongly influenced by the coupling between solute and solvent degrees of freedom. The motion of a solute particle creates a perturbation of the solvent that spreads out to other particles. Hence, the particle movements become correlated due to viscous momentum diffusion. The idea of hybrid fluid-particle methods is to simulate the momentum transport in an explicit solvent with the LBM while the solute objects are coupled to the flow field by some sort of interaction force [9]. If the immersed objects have internal degrees of freedom, e.g. polymers, membranes or cells, the equations of motion for these degrees of freedom have to be solved simultaneously with the fluid dynamics. Various approaches have been proposed to solve this task, such as the immersed boundary method (IBM) [20], force coupling through viscous drag [13], or external boundary force (EBF) [18]. They have in common that a set of Lagrangian particles or marker points is added to the lattice Boltzmann fluid, which require the simultaneous integration of Newton’s equations of motion, e.g., by molecular dynamics (MD) techniques.

Integration schemes for MD can be faced with spurious effects due to the time discretization, for example, systematic drift of energy or temperature, artificial spatial correlations between particles [21], and numerical instabilities [22]. A number of improved algorithms has been introduced, e.g. [21, 23–25], which are commonly based on a Trotter expansion of the Liouville propagator [26–28]. Integration schemes derived by splitting methods can improve accuracy and lead to symplectic algorithms that are usually also more stable [23, 29, 30].

In this communication, a general approach to derive time-marching schemes to solve hybrid fluid-particle systems is presented. It is based on the interpretation of the lattice Boltzmann equation as a Strang splitting scheme [31] combined with common operator splitting techniques used in molecular dynamics (MD) and related methods [23, 30, 32–35]. The operator splitting makes

it straightforward to derive explicit time-discretizations, in particular, second-order time-accurate integration schemes for the various coupling methods are proposed. Furthermore, a modified external boundary force is proposed that ensures the no-slip boundary condition after application of the force operator. The latter also reveals that the IBM can be interpreted as the limit of force coupling methods for vanishing particle mass, and that the force transfer during coupling is controlled by the ratio between fluid and particle mass. This notion may also be applicable to other mesoscopic methods such as multi-particle collision dynamics [36, 37].

The remainder of this work is organized as follows: In section 2 and 3, the derivation of the lattice Boltzmann equation and its interpretation as an operator splitting scheme [31] are briefly reviewed. In section 4, an alternative lattice Boltzmann algorithm in the form of a collide-stream-collide scheme is outlined and analyzed. The operator splitting is then extended to several coupling methods, i.e., the immersed boundary method (section 5.1), force coupling (section 5.2), and external boundary force (section 5.3). Finally, multiple time-step schemes are briefly discussed in section 6.

2. Discrete Boltzmann equation

In the kinetic theory of gases, the dynamics and transport properties of fluids are determined by the Boltzmann equation [38]

$$\left(\frac{\partial}{\partial t} + \mathbf{c} \cdot \frac{\partial}{\partial \mathbf{x}} + \mathbf{a} \cdot \frac{\partial}{\partial \mathbf{c}} \right) f(\mathbf{x}, \mathbf{c}, t) = -\Omega(f - f^{\text{eq}}). \quad (1)$$

Here, $f(\mathbf{x}, \mathbf{c}, t)$ is the one-particle distribution function at position $\mathbf{x}$, velocity $\mathbf{v}$, and time t , and the collision operator Ω on the right-hand side has been linearized around the equilibrium distribution f^{eq} [39, 40]. In order to discretize the Boltzmann equation (1), the moment integrals of the distribution function are formally evaluated by a Gauss-Hermite quadrature, where the abscissae of the quadrature comprise a set of discrete velocities $\mathbf{c}_i$ [4, 41]. This procedure leads to the discrete Boltzmann equation, which is also the basis of discrete velocity models [42, 43]

$$\left(\frac{\partial}{\partial t} + \mathbf{c}_i \cdot \frac{\partial}{\partial \mathbf{x}} \right) f_i = - \sum_j \Omega_{ij} (f_j - f_j^{\text{eq}}) + G_i, \quad (2)$$

where G_i is a body force corresponding to the acceleration term $-\mathbf{a} \cdot \nabla_{\mathbf{c}} f$. The standard approach to space-time discretization of the discrete Boltzmann equation is integrating along the characteristic $(\mathbf{x} + h\mathbf{c}_i, t + h)$ and approximating the integral of the collision term by the trapezium rule [2, 3]. This leads to a non-linear implicit equation for $f_i(\mathbf{x} + h\mathbf{c}_i, t + h)$ that can be turned into an explicit scheme by an appropriate re-definition of variables [44]. This explicit scheme is the lattice Boltzmann equation in the standard form [45–48]

$$\tilde{f}_i(\mathbf{x} + h\mathbf{c}_i, t + h) = \tilde{f}_i(\mathbf{x}, t) - \sum_j \Lambda_{ij} \left(\tilde{f}_j - f_j^{\text{eq}} \right) + \sum_j \left(\delta_{ij} - \frac{1}{2} \Lambda_{ij} \right) h G_j(t), \quad (3)$$

where the new variables are [44, 49]

$$\tilde{f}_i(t) = f_i(t) + \sum_j \frac{h}{2} \Omega_{ij} (f_j - f_j^{\text{eq}}) - \frac{h}{2} G_i(t). \quad (4)$$

It is important to make the distinction between f_i and the new variables $\tilde{f}_i$ and to replace the collision matrix $h\Omega$ by the discrete collision matrix $\Lambda = (1 + \frac{h}{2}\Omega)^{-1}h\Omega$ [49–51]. Without these replacements, Eq. (3) is only a first-order accurate discretization of the Boltzmann equation [52]. While Λ enters in the collision step and leads to the correct viscosity, the variable transformation (4) is essential to evaluate the macroscopic flow fields which have to be obtained as moments of the back-transformed variables f_i :

$$\rho = \sum_i f_i, \quad \rho \mathbf{u} = \sum_i f_i \mathbf{c}_i, \quad \Pi = \sum_i f_i \mathbf{c}_i \mathbf{c}_i. \quad (5)$$

In the absence of forces, the back-transformation is only relevant for evaluating the (non-equilibrium) stress and does not change the mass and momentum densities.

Dellar [31] has recently pointed out that the lattice Boltzmann equation can also be interpreted as a Strang splitting scheme. In the following, operator splitting for the discrete Boltzmann equation is revisited, which will serve as the basis to derive accurate coupling schemes in later sections. The operator splitting approach will also provide a complementary view on the role of Eq. (4).

3. Operator splitting for the discrete Boltzmann equation

In the operator splitting interpretation put forward by Dellar [31], the lattice Boltzmann equation is split into decoupled streaming, collision and forcing steps. The idea is to find approximate solutions to the individual steps and combine them into a time-marching scheme, the accuracy of which depends on the order of the applied splitting and the accuracy of the individual approximate solutions. Splitting methods [29] make use of decomposition formulas for exponential operators where the exponential of a sum of non-commuting operators is approximated by a symmetrized product of exponentials of the individual operators [26–28]. For the simplest case of two operators, the resulting splitting scheme is equivalent to the difference scheme introduced by Strang [53]. The decomposition formulas are valid for linear operators and can be generalized to nonlinear operators by application of the Baker-Campbell-Hausdorff formula [54–56]. The truncated Taylor series of the exponentials can be applied to sufficiently differentiable functions and yield formally accurate splitting schemes [56, 57]. Hence it is justified to use splitting schemes for the Boltzmann kinetic equation [58] although the collision operator depends non-linearly on the flow velocity through the equilibrium distribution. The remainder of this section closely follows the work of Dellar [31]. However, instead of combining adjacent parts of different time-steps to recover the standard lattice Boltzmann algorithm up to a transformation of variables, we will combine the sub-parts of one

time-step into an alternative update scheme in section 4. This provides a complementary view on the splitting scheme and clarifies the role of the transformed variables as intermediate values.

The discrete Boltzmann equation (2) can be written in terms of operators as

$$\frac{\partial}{\partial t} f_i(\mathbf{x}, t) = (\mathcal{S} + \mathcal{C} + \mathcal{F}) f_i(\mathbf{x}, t), \quad (6)$$

where the operators for streaming $\mathcal{S}$, collisions $\mathcal{C}$, and forces $\mathcal{F}$ are

$$\mathcal{S} f_i = -\mathbf{c}_i \cdot \frac{\partial}{\partial \mathbf{x}} f_i, \quad (7a)$$

$$\mathcal{C} f_i = - \sum_j \Omega_{ij} (f_j - f_j^{\text{eq}}), \quad (7b)$$

$$\mathcal{F} f_i = G_i. \quad (7c)$$

The action of forces is usually incorporated into the collision operator $\mathcal{C}$, but it can also be treated as a third decoupled step [31, 59, 60]. Here, we adopt the splitting using three decoupled parts for streaming, collisions and forces. The motivation is that the force operator couples the interactions with the lattice Boltzmann equation and may be implemented more cleanly if treated separately.

The formal solutions of the decoupled equations (7) are

$$f_i(\mathbf{x}, t+h) = e^{h\mathcal{S}} f_i(\mathbf{x}, t) \approx \mathbf{S} f_i(\mathbf{x}, t), \quad (8a)$$

$$f_i(\mathbf{x}, t+h) = e^{h\mathcal{C}} f_i(\mathbf{x}, t) \approx \mathbf{C} f_i(\mathbf{x}, t), \quad (8b)$$

$$f_i(\mathbf{x}, t+h) = e^{h\mathcal{F}} f_i(\mathbf{x}, t) \approx \mathbf{F} f_i(\mathbf{x}, t), \quad (8c)$$

where the three operators $\mathbf{S} \approx \exp[h\mathcal{S}]$, $\mathbf{C} \approx \exp[h\mathcal{C}]$, and $\mathbf{F} \approx \exp[h\mathcal{F}]$ are introduced as approximations to the separate exponential operators. Note that the index i and the coordinate $\mathbf{x}$ in the streaming part (8a) are to be understood as ranging over all possible values because streaming is a non-local operation. In this sense, the operator $\exp[h\mathcal{S}]$ acts on the whole ensemble of distributions. The discrete approximations for the decoupled steps are obtained below.

The formal solution to (6) is given by

$$f_i(\mathbf{x}, t+h) = e^{h(\mathcal{S}+\mathcal{C}+\mathcal{F})} f_i(\mathbf{x}, t), \quad (9)$$

where again the index i and the coordinates $\mathbf{x}$ are ranging over all values. A second-order Trotter-Suzuki decomposition [26, 28] of the exponential operator leads to the splitting scheme

$$\begin{aligned} e^{h(\mathcal{S}+\mathcal{C}+\mathcal{F})} &= e^{\frac{h}{2}\mathcal{F}} e^{\frac{h}{2}\mathcal{C}} e^{h\mathcal{S}} e^{\frac{h}{2}\mathcal{C}} e^{\frac{h}{2}\mathcal{F}} + O(h^3), \\ &= \mathbf{F}^{\frac{1}{2}} \mathbf{C}^{\frac{1}{2}} \mathbf{S} \mathbf{C}^{\frac{1}{2}} \mathbf{F}^{\frac{1}{2}} + O(h^3), \end{aligned} \quad (10)$$

where the operators $\mathbf{S}$, $\mathbf{C}^{\frac{1}{2}}$, and $\mathbf{F}^{\frac{1}{2}}$ are the discrete approximations of the exponential operators for a step h and $h/2$, respectively. The lattice Boltzmann algorithm requires that the streaming part $\mathbf{S}$ must not be split in order to be

compatible with the lattice structure. For second-order accuracy, $F^{\frac{1}{2}}$ and $C^{\frac{1}{2}}$ have to be arranged symmetrically but their order is in principle arbitrary. The choice to put $F^{\frac{1}{2}}$ to the outside is motivated by the coupling algorithms discussed below. For small enough forces $F^{\frac{1}{2}}$ and $C^{\frac{1}{2}}$ commute, cf. section 3.3.

3.1. Streaming

The streaming part (8a) can be written as a derivative along the characteristic $(x + h'\mathbf{c}_i, t + h')$

$$\left(\frac{\partial}{\partial t} + \mathbf{c}_i \cdot \frac{\partial}{\partial \mathbf{x}}\right) f_i(\mathbf{x}, t) = \frac{d}{dh'} f_i(\mathbf{x} + h'\mathbf{c}_i, t + h') = 0, \quad (11)$$

and can be integrated from 0 to h to give

$$\int_0^h \frac{d}{dh'} f_i(\mathbf{x} + h'\mathbf{c}_i, t + h') dh' = f_i(\mathbf{x} + h\mathbf{c}_i, t + h) - f_i(\mathbf{x}, t) = 0. \quad (12)$$

We thus get the discrete streaming step

$$f_i(\mathbf{x} + h\mathbf{c}_i, t + h) = f_i(\mathbf{x}, t), \quad (13)$$

from which the spatial grid structure emerges, i.e., only coordinates connected by the discrete vectors $h\mathbf{c}_i$ are admissible.

3.2. Collisions

The collision part (8b) leaves the equilibrium distribution invariant since $\mathcal{C}f_i^{\text{eq}} = 0$. Hence we consider only the non-equilibrium part $f_i^{\text{neq}} = f_i - f_i^{\text{eq}}$ and the differential equation

$$\frac{d}{dt} \mathbf{f}^{\text{neq}} = \mathcal{C} \mathbf{f}^{\text{neq}} = -\Omega \mathbf{f}^{\text{neq}}, \quad (14)$$

where we have introduced the vector notation $\mathbf{f} = (f_0, f_1, \dots, f_q)^T$. An approximate solution is obtained using the $O(h^3)$ accurate Crank-Nicolson rule [31, 49]

$$\mathbf{f}^{\text{neq}}(t + h) - \mathbf{f}^{\text{neq}}(t) = -\frac{h}{2} \Omega [\mathbf{f}^{\text{neq}}(t + h) + \mathbf{f}^{\text{neq}}(t)], \quad (15)$$

which leads to

$$\mathbf{f}^{\text{neq}}(t + h) = \left(1 + \frac{h}{2} \Omega\right)^{-1} \left(1 - \frac{h}{2} \Omega\right) \mathbf{f}^{\text{neq}}(t). \quad (16)$$

A comparison with the exact solution

$$\mathbf{f}^{\text{neq}}(t + h) = \exp[-h\Omega] \mathbf{f}^{\text{neq}}(t) \quad (17)$$

shows that the accuracy of (16) is in fact $O((h/\tau)^3)$, which can cause nonlinear instabilities for grid scale Reynolds number h/τ above unity [19]. On the other

hand, the analysis of Dellar [31] revealed that the exact solution (17) leads to excessively rapid decay of shear modes as $\tau \lesssim h$, while the Crank-Nicolson approximation (16) gives accurate decay rates even for small collision times. The reason for this somewhat counter-intuitive result lies in the compensation of errors introduced by the splitting of the Boltzmann equation into uncoupled steps on one hand, and the Crank-Nicolson approximation of the collision operator on the other.

Equation (16) can be rewritten as

$$\begin{aligned} \mathbf{C}\mathbf{f}^{\text{neq}} &= \left[\mathbf{I} - \left(\mathbf{I} + \frac{h}{2}\Omega \right)^{-1} h\Omega \right] \mathbf{f}^{\text{neq}} \\ &= (\mathbf{I} - \Lambda) \mathbf{f}^{\text{neq}} \end{aligned} \quad (18)$$

with the discrete collision matrix [49]

$$\Lambda = \left(\mathbf{I} + \frac{h}{2}\Omega \right)^{-1} h\Omega. \quad (19)$$

This corresponds to the familiar replacement of the collision time τ by $\tau + h/2$ in the discrete scheme [50]. The half-step collision update $\mathbf{C}^{\frac{1}{2}}$ in the splitting scheme (10) is obtained by replacing h by $h/2$ in Eq. (19)

$$\Lambda_{\frac{1}{2}} = \left(\mathbf{I} + \frac{h}{4}\Omega \right)^{-1} \frac{h}{2}\Omega. \quad (20)$$

We can convince ourselves that with Λ and $\Lambda_{\frac{1}{2}}$ as in (19) and (20), two applications of the half-step collisions $\mathbf{C}^{\frac{1}{2}}$ are up to order $O(h^3)$ equivalent to one full-step collision $\mathbf{C}$

$$\begin{aligned} \mathbf{C}^{\frac{1}{2}}\mathbf{C}^{\frac{1}{2}}\mathbf{f}^{\text{neq}} &= (\mathbf{I} - \Lambda_{\frac{1}{2}})^2 \mathbf{f}^{\text{neq}} \\ &= (\mathbf{I} - \Lambda) \mathbf{f}^{\text{neq}} + O(h^3) \\ &= \mathbf{C}\mathbf{f}^{\text{neq}} + O(h^3). \end{aligned} \quad (21)$$

The order of accuracy follows from $(\mathbf{I} - \Lambda_{\frac{1}{2}})^2 = (\mathbf{I} - \Lambda) + O(h^3)$. Given Λ , we can also exactly define $\mathbf{I} - \Lambda_{\frac{1}{2}} = \sqrt{\mathbf{I} - \Lambda}$, where the square-root is to be understood in the sense of a Cholesky decomposition. For the eigenvalues, we simply have $1 - \lambda_{\frac{1}{2}} = \sqrt{1 - \lambda}$. In contrast, the approximation $\mathbf{C}^{\frac{1}{2}} = \frac{1}{2}(\mathbf{I} + \mathbf{C})$ is only $O(h^2)$ accurate since $\Lambda_{\frac{1}{2}} = \Lambda/2 + O(h^2)$

$$\begin{aligned} \mathbf{C}^{\frac{1}{2}}\mathbf{f}^{\text{neq}} &= \left(\mathbf{I} - \Lambda_{\frac{1}{2}} \right) \mathbf{f}^{\text{neq}} \\ &= \left(\mathbf{I} - \frac{1}{2}\Lambda \right) \mathbf{f}^{\text{neq}} + O(h^2) = \frac{1}{2}(\mathbf{I} + \mathbf{C})\mathbf{f}^{\text{neq}} + O(h^2). \end{aligned} \quad (22)$$

When the half-step collisions are applied in every iteration of the time-marching scheme, the accuracy of $\mathbf{C}^{\frac{1}{2}}$ is essential to maintain the global order $O(h^2)$, cf. section 4.

For the sake of completeness, we mention that in the multi-relaxation time approach (MRT), the collisions are typically implemented in the space of moments m_k of the f_i :

$$\mathbf{C} m_k^{\text{neq}} = (1 - \lambda_k) [m_k - m_k^{\text{eq}}]. \quad (23)$$

Here, λ_k is the k -th eigenvalue of the collision matrix which is diagonal in moment space. In the fluctuating lattice Boltzmann method [11, 61, 62], a noise term is added to the collision step

$$\mathbf{C} m_k^{\text{neq}} = (1 - \lambda_k) [m_k - m_k^{\text{eq}}] + \sqrt{\rho} \varphi_k r_k, \quad (24)$$

where r_k is a Gaussian random number, and the amplitude $\varphi_k = \sqrt{\mu b_k \lambda_k^2}$ is related to the eigenvalue λ_k and the temperature through $\mu = a^{-3} k_b T / c_s^2$. b_k is the length of the k -th basis vector of the specific lattice model. Equations (23) and (24) are the basis for the implementation of the collision step in Algorithm 1. It is worth noting that the fluctuating collision operator can be split into half-steps by a proper choice of the eigenvalues and noise strength. A backwards half-step application $\mathbf{C}^{-\frac{1}{2}}$ as needed below in equation (30), however, is not straightforward to perform without distorting the correlations of the moments.

3.3. Forces

The force operator $\mathbf{F}$ in Eq. (8c) can be discretized using the midpoint rule

$$\mathbf{f}(t+h) = \mathbf{F} \mathbf{f}(t) = \mathbf{f}(t) + h \mathbf{G}(t + \frac{h}{2}), \quad (25a)$$

or alternatively with the Crank-Nicolson rule

$$\mathbf{f}(t+h) = \mathbf{F} \mathbf{f}(t) = \mathbf{f}(t) + h \frac{\mathbf{G}(t+h) + \mathbf{G}(t)}{2}. \quad (25b)$$

The force term $\mathbf{G} = (G_1, G_2, \dots, G_q)^T$ has the same first moments as the acceleration term $-\mathbf{a} \cdot \nabla_{\mathbf{c}} f$ [48, 63]

$$\sum_i G_i = 0, \quad \sum_i \mathbf{c}_i G_i = \mathbf{F}, \quad \sum_i \mathbf{c}_i \mathbf{c}_i G_i = \mathbf{F} \mathbf{u} + \mathbf{u} \mathbf{F}, \quad (26)$$

where $\mathbf{F} = \rho \mathbf{a}$. Note that the minus sign for bringing $\mathbf{a} \cdot \nabla_{\mathbf{c}} f$ to the right hand side is compensated by another minus sign that arises in the Hermite expansion of $\nabla_{\mathbf{c}} f$ [64]. In terms of the first moments, Eqs. (25a) and (25b) hence read

$$\begin{aligned} \mathbf{u}(t+h) &= \mathbf{u}(t) + \frac{h}{\rho} \mathbf{F}(t + \frac{h}{2}), \\ \Pi(t+h) &= \Pi(t) + h \left[\mathbf{u}(t + \frac{h}{2}) \mathbf{F}(t + \frac{h}{2}) + \mathbf{F}(t + \frac{h}{2}) \mathbf{u}(t + \frac{h}{2}) \right], \end{aligned} \quad (\text{midpoint}) \quad (27a)$$

and

$$\begin{aligned}
\mathbf{u}(t+h) &= \mathbf{u}(t) + \frac{h}{2\rho} (\mathbf{F}(t+h) + \mathbf{F}(t)), \\
\Pi(t+h) &= \Pi(t) + \frac{h}{2} [\mathbf{u}(t)\mathbf{F}(t) + \mathbf{F}(t)\mathbf{u}(t) \\
&\quad + \mathbf{u}(t+h)\mathbf{F}(t+h) + \mathbf{F}(t+h)\mathbf{u}(t+h)].
\end{aligned}
\tag{Crank-Nicolson}
\tag{27b}$$

If the force is constant during a time step h , Crank-Nicolson and midpoint are both equivalent to forward Euler discretization, where the midpoint velocity $\mathbf{u}(t+h/2)$ has to be used in the stress update.

It is worthwhile to note that up to $O(\mathbf{F}^2)$, the force operator $\mathbf{F}$ only acts on the equilibrium distribution and leaves the non-equilibrium part unchanged

$$\begin{aligned}
\mathbf{f}(t+h) - \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t+h)) &= \mathbf{f}(t) + h\mathbf{G}(t + \frac{h}{2}) - \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t+h)) \\
&= \mathbf{f}(t) - \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t)) + O(\mathbf{F}^2),
\end{aligned}
\tag{28}$$

which can easily be checked using the truncated Hermite expansion of the force term G_i and the equilibrium distribution f_i^{eq} [48, 63, 64]. As a consequence, the force and collision operators approximately commute

$$\begin{aligned}
\mathbf{C}\mathbf{F}\mathbf{f}(t) &= \mathbf{F}\mathbf{f}(t) - \Lambda\mathbf{F}[\mathbf{f}(t) - \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t))] \\
&= \mathbf{f}(t) + h\mathbf{G} - \Lambda[\mathbf{f}(t) - \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t))] + O(\mathbf{F}^2) \\
&= \mathbf{C}\mathbf{f}(t) + h\mathbf{G} + O(\mathbf{F}^2) \\
&= \mathbf{F}\mathbf{C}\mathbf{f}(t) + O(\mathbf{F}^2).
\end{aligned}
\tag{29}$$

For small enough forces the order of separate collisions and forcing thus does not matter. Note that throughout this section we have discussed a full time step. The action of the half-step force operator $\mathbf{F}^{\frac{1}{2}}$ is obtained by simply replacing h by $h/2$. We will see below that the splitting scheme leads to the familiar transformation of the force term required for second-order accuracy [47, 48, 63].

4. Collide-stream-collide lattice Boltzmann

Strang or Trotter-Suzuki splitting as in Eq. (10) leads to a sequence of operations which involve the half-step collision operator $\mathbf{C}^{\frac{1}{2}}$. If it is the first and last operation, then upon concatenation of time-steps the adjacent collision half-steps can be combined, and the standard LB algorithm is recovered [31]

$$\left[\mathbf{C}^{\frac{1}{2}} \mathbf{S} \mathbf{C}^{\frac{1}{2}} \right]^n = \mathbf{C}^{\frac{1}{2}} [\mathbf{S}\mathbf{C}]^n \mathbf{C}^{-\frac{1}{2}},
\tag{30}$$

where the forces are included in the collision operator. At the beginning and end of the simulation the half-step collisions $\mathbf{C}^{-\frac{1}{2}}$ and $\mathbf{C}^{\frac{1}{2}}$ have to be applied,

respectively. In this work, we group the operations such that the force operator $\mathbf{F}$ does not interrupt the collision and streaming steps, which facilitates the implementation of hybrid coupling algorithms with sub-splitting of $\mathbf{F}$ (cf. section 6).

Before developing the details of the coupling, we analyze the splitting $\mathbf{C}^{\frac{1}{2}}\mathbf{S}\mathbf{C}^{\frac{1}{2}}$ for the LB part. The latter suggests a collide-stream-collide lattice Boltzmann update with two half-step collisions before and after the streaming. Let f_i^* denote the populations after the first collision step, and $\bar{f}_i$ the populations after the streaming step. Then the three separate steps are

$$\mathbf{f}(\mathbf{x}, t) = \bar{\mathbf{f}}(\mathbf{x}, t) - \Lambda_{\frac{1}{2}} [\bar{\mathbf{f}}(\mathbf{x}, t) - \mathbf{f}^{\text{eq}}(\mathbf{x}, t)] \quad (31a)$$

$$\bar{f}_i(\mathbf{x} + h\mathbf{c}_i, t + h) = f_i^*(\mathbf{x}, t) \quad (31b)$$

$$\mathbf{f}^*(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - \Lambda_{\frac{1}{2}} [\mathbf{f}(\mathbf{x}, t) - \mathbf{f}^{\text{eq}}(\mathbf{x}, t)]. \quad (31c)$$

The difference to a direct Crank-Nicolson discretization of the Boltzmann equation is that two explicit collisions are applied, and not just one collision using a midpoint approximation for the non-equilibrium distribution [51, 65]. To see the relation to the standard LB scheme, we note that equation (31b) for the post-streaming distribution can be rewritten as

$$\begin{aligned} \bar{f}_i(\mathbf{x} + h\mathbf{c}_i, t + h) &= f_i^*(\mathbf{x}, t) \\ &= f_i^{\text{eq}}(\mathbf{x}, t) + \sum_j (1 - \Lambda_{\frac{1}{2}})_{ij} [f_j(\mathbf{x}, t) - f_j^{\text{eq}}(\mathbf{x}, t)] \\ &= f_i^{\text{eq}}(\mathbf{x}, t) + \sum_j (1 - \Lambda_{\frac{1}{2}})_{ij}^2 [\bar{f}_j(\mathbf{x}, t) - f_j^{\text{eq}}(\mathbf{x}, t)], \end{aligned} \quad (32)$$

which is the standard LB scheme with collision matrix $(1 - \Lambda_{\frac{1}{2}})^2$. A Chapman-Enskog expansion [66] reveals that the moments of $\bar{f}_i$ satisfy the Navier-Stokes equation, which allows us to obtain explicit expressions for the transport coefficients.

The viscosity ν is given by [9, 12, 67, 68]

$$\nu = \frac{hc_s^2}{2} \frac{1 + \gamma_{\frac{1}{2}}^2}{1 - \gamma_{\frac{1}{2}}^2} = hc_s^2 \left(\frac{1}{2\lambda_{\frac{1}{2}} - \lambda_{\frac{1}{2}}^2} - \frac{1}{2} \right) = hc_s^2 \left(\frac{1}{2\lambda_{\frac{1}{2}}} - \frac{1}{4} \right) + O(h^2), \quad (33)$$

where $\gamma_{\frac{1}{2}} = 1 - \lambda_{\frac{1}{2}}$ is the eigenvalue of $(1 - \Lambda_{\frac{1}{2}})$ corresponding to the shear modes. The relation to the continuous Boltzmann equation with BGK collisions [39] is achieved by shifting the BGK relaxation time τ according to the time discretization [50, 51, 65], cf. Eq. (20),

$$\lambda_{\frac{1}{2}} = \frac{h}{2} \frac{1}{\tau + \frac{h}{4}}, \quad (34)$$

such that

$$\nu = hc_s^2 \left(\frac{\tau + \frac{h}{4}}{h} - \frac{1}{4} \right) = \tau c_s^2, \quad (35)$$

in accordance with the continuum result [66].

The viscous stress σ is given by [9, 68]

$$\begin{aligned}
\sigma &= \frac{1}{2} \left(1 + (1 - \Lambda_{\frac{1}{2}})^2 \right) \sum_i (\bar{f}_i - f_i^{\text{eq}}) \mathbf{c}_i \mathbf{c}_i \\
&= \frac{(1 - \Lambda_{\frac{1}{2}})^{-1} + (1 - \Lambda_{\frac{1}{2}})}{2} \sum_i (f_i - f_i^{\text{eq}}) \mathbf{c}_i \mathbf{c}_i \\
&= \sum_i (f_i - f_i^{\text{eq}}) \mathbf{c}_i \mathbf{c}_i + O(h^2).
\end{aligned} \tag{36}$$

The second line corresponds to a midpoint scheme that averages the stress before the last half-step collision and after the next half-step collision. The last line shows that in the collide-stream-collide scheme, the viscous stress is equal to the local non-equilibrium stress up to $O(h^2)$ and can directly be obtained from the distributions f_i . No additional transformation or look-ahead collision as in the standard scheme has to be performed.

If we approximate $\mathbf{C}^{\frac{1}{2}} = \frac{1}{2}(1 + \mathbf{C})$, the half-step collisions are given by (22) and read

$$\mathbf{C}^{\frac{1}{2}} \mathbf{f}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - \left(1 + \frac{h}{2} \Omega \right)^{-1} \frac{h}{2} \Omega [\mathbf{f}(\mathbf{x}, t) - \mathbf{f}^{\text{eq}}(\mathbf{x}, t)] + O(h^2). \tag{37}$$

In this approximation, as pointed out by Dellar [31], the second half-step collision (31a) coincides with the transformation introduced by He et al. [44]. This demonstrates that the different interpretations of the operator splitting lead to equivalent LB schemes. More precisely, the LB variables $\bar{f}_i$ in (3) correspond in fact to the intermediate post-streaming distributions $\tilde{f}_i$ in the splitting scheme, and the transformation (31a) is necessary to complete the full time-step. It is to be noted that the approximation in Eq. (22) is insufficient for global second order accuracy if the half-step collisions are applied in every step, since the error is $O(h^2)$ per step. Hence, the choice $\lambda_{\frac{1}{2}} = \lambda/2$ does not lead to the same viscosity as the standard scheme, which can be easily seen from Eq. (33). The difference can be traced back to the shift $h/4$ in Eq. (34) which is a consequence of the splitting of the collisions into two half-steps. The correct relaxation parameter $\lambda_{\frac{1}{2}}$ can instead be directly determined from Eq. (33) or (34). It should also be noted that it is not straightforward to invert (37) if the collision operator includes fluctuations.

The force operator $\mathbf{F}$ is readily included in the splitting scheme $\mathbf{F}^{\frac{1}{2}} \mathbf{C}^{\frac{1}{2}} \mathbf{S} \mathbf{C}^{\frac{1}{2}} \mathbf{F}^{\frac{1}{2}}$.

Equations (31) for the intermediate steps become

$$\mathbf{f}(\mathbf{x}, t) = \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t - \frac{h}{2})) + (\mathbf{I} - \Lambda_{\frac{1}{2}}) \left[\bar{\mathbf{f}}(\mathbf{x}, t) - \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t - \frac{h}{2})) \right] + \frac{h}{2} \mathbf{G}(t - \frac{h}{4}) \quad (38a)$$

$$\bar{f}_i(\mathbf{x} + h\mathbf{c}_i, t + h) = f_i^*(\mathbf{x}, t) \quad (38b)$$

$$\mathbf{f}^*(\mathbf{x}, t) = \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t)) + (\mathbf{I} - \Lambda_{\frac{1}{2}}) [\mathbf{f}(\mathbf{x}, t) - \mathbf{f}^{\text{eq}}(\rho, \mathbf{u}(t))] + \frac{h}{2} \mathbf{G}(t + \frac{h}{4}), \quad (38c)$$

where we have used midpoint forces in both forcing steps. As before, the relation to the standard LB scheme can be elucidated in terms of the post-streaming variables $\bar{f}_i(\mathbf{x} + h\mathbf{c}_i, t + h)$. Inserting (38a) and (38c) into (38b) we obtain

$$\begin{aligned} \bar{f}_i(\mathbf{x} + h\mathbf{c}, t + h) &= f_i^{\text{eq}}(\rho, \mathbf{u}(t)) + \sum_j (\mathbf{I} - \Lambda_{\frac{1}{2}})_{ij} [f_j(\mathbf{x}, t) - f_j^{\text{eq}}(\rho, \mathbf{u}(t))] + \frac{h}{2} G_i(t + \frac{h}{4}) \\ &= f_i^{\text{eq}}(\rho, \mathbf{u}(t)) + \sum_j (\mathbf{I} - \Lambda_{\frac{1}{2}})_{ij}^2 \left[\bar{f}_j(\mathbf{x}, t) - f_j^{\text{eq}}(\rho, \mathbf{u}(t - \frac{h}{2})) \right] + \frac{h}{2} G_i(t + \frac{h}{4}) \\ &= f_i^{\text{eq}}(\rho, \mathbf{u}(t)) + \sum_j (\mathbf{I} - \Lambda_{\frac{1}{2}})_{ij}^2 [\bar{f}_j(\mathbf{x}, t) - f_j^{\text{eq}}(\rho, \mathbf{u}(t))] \\ &\quad + \sum_j (\mathbf{I} - \Lambda_{\frac{1}{2}})_{ij}^2 \frac{h}{2} G_j(t - \frac{h}{4}) + \frac{h}{2} G_i(t + \frac{h}{4}). \end{aligned} \quad (39)$$

Assuming that the force term G_i can be expanded around t , we find up to terms of $O(h^3)$

$$\begin{aligned} \bar{f}_i(\mathbf{x} + h\mathbf{c}_i, t + h) &= f_i^{\text{eq}}(\rho, \mathbf{u}(t)) + \sum_j (\mathbf{I} - \Lambda_{\frac{1}{2}})_{ij}^2 [\bar{f}_j(\mathbf{x}, t) - f_j^{\text{eq}}(\rho, \mathbf{u}(t))] \\ &\quad + \sum_j \left(\mathbf{I} + (\mathbf{I} - \Lambda_{\frac{1}{2}})^2 \right)_{ij} \frac{h}{2} G_j(t) + O(h^3) \\ &= f_i^{\text{eq}}(\rho, \mathbf{u}(t)) + \sum_j (\mathbf{I} - \Lambda)_{ij} [\bar{f}_j(\mathbf{x}, t) - f_j^{\text{eq}}(\rho, \mathbf{u}(t))] \\ &\quad + \sum_j \left(\mathbf{I} - \frac{1}{2} \Lambda \right)_{ij} h G_j(t) + O(h^3). \end{aligned} \quad (40)$$

Equation (40) contains the familiar transformation of the force term $\mathbf{G}$ that is used in standard LB schemes to maintain second-order accuracy [9, 12, 47, 48]. In the splitting scheme it arises naturally without the need to know the transformation a-priori.

Another aspect that is illuminated by the derivation is the redefinition of the hydrodynamic velocity in the presence of external forces [9, 12]

$$\rho \mathbf{u} = \sum_i \bar{f}_i \mathbf{c}_i \mathbf{c}_i + \frac{h}{2} \mathbf{F}, \quad (41)$$

which is simply a consequence of the second application of the force operator $\mathbf{F}^{\frac{1}{2}}$ after the streaming step.

Although here the primary purpose of the collide-stream-collide scheme is to elucidate the operator splitting approach, we remark that it is straightforward to implement using a combination of the familiar “push” and “pull” schemes [69]. A simple pseudo-code is listed in Algorithm 1. If one uses a memory-efficient data layout for intermediate results, there is only a small overhead in form of the floating point operations for the second relaxation step. This may be almost negligible because the performance of the LBM on modern computing architectures is mainly limited by memory bandwidth, and the bottleneck regarding lattice site updates per second is really the streaming step. Nevertheless, it may seem that in terms of floating point operations, the collide-stream-collide LB scheme is less efficient due to the extra collision step. On the other hand, the standard LB scheme requires extra effort to evaluate the viscous stress, cf. Eq. (36). As long as the stress is only needed at the end of the calculation, the standard LB scheme may be preferable. However, if a coupling scheme involves the fluid stress, its computation is needed after every update. In this case the collide-stream-collide is almost equally efficient, and it has the additional advantage that the viscous stress is directly available as the local non-equilibrium stress.

5. Coupling methods for hybrid fluid-particle systems

We now turn to methods for coupling immersed particles to the lattice Boltzmann equation that allow to simulate complex fluids. Here, we focus on objects with internal degrees of freedom, such as polymers or vesicles. Unlike colloids, they can not easily be coupled to the fluid by simple boundary conditions due to their complex and changing shape. The most common methods employed in such cases are the immersed boundary method (IBM), force coupling (FC), and external boundary force (EBF).

5.1. Immersed boundary method (IBM)

In the immersed boundary method [20], a fluid-object interface is represented by a set of Lagrangian nodes with positions $\mathbf{X}_i$ that are advected with the flow. Interactions on the Lagrangian grid are applied as body forces to the fluid. The positions of the Lagrangian nodes are additional degrees of freedom that are updated according to

$$\frac{d}{dt} \mathbf{X}_i(t) = \mathcal{P} \mathbf{X}_i(t) = \mathbf{u}(\mathbf{X}_i, t), \quad (42)$$

where $\mathcal{P}$ is the propagation operator and $\mathbf{u}(\mathbf{X}_i, t)$ is the flow velocity at the Lagrangian position $\mathbf{X}_i$. The latter requires a mapping between Lagrangian positions and Eulerian grid coordinates

$$\mathbf{u}(\mathbf{X}_i, t) = \mathcal{I}_a[\mathbf{X}_i(t)] \mathbf{u}(\mathbf{X}_i) = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x}, t) \delta_a(\mathbf{X}_i(t) - \mathbf{x}), \quad (43)$$

where δ_a is a discrete Dirac delta function on a lattice with lattice spacing a , and $\mathcal{I}_a[\mathbf{X}_i]$ is an interpolation operator based on the Lagrangian positions $\mathbf{X}_i$. Conversely, the forces arising between the Lagrangian nodes are spread to the grid points with the adjoint operator $\mathcal{I}_a^*$

$$\mathbf{F}(\mathbf{x}, t) = \mathcal{I}_a^*[\mathbf{X}_i(t)] \mathbf{F}_i(\mathbf{x}) = \frac{1}{a^3} \sum_i \mathbf{F}_i(t) \delta_a(\mathbf{X}_i(t) - \mathbf{x}). \quad (44)$$

Note that $\mathbf{F}(\mathbf{x}, t)$ is a force density while $\mathbf{F}_i(t)$ is a force, hence the factor a^{-3} on the right hand side of (44). The construction of the discrete delta function δ_a is described in detail in [20] and is briefly summarized in Appendix A.

The evolution of the complete system is described by

$$\frac{\partial}{\partial t} \{\mathbf{f}(\mathbf{x}, t), \mathbf{X}_i(t)\} = [\mathcal{S} + \mathcal{C} + \mathcal{F} + \mathcal{P}] \{\mathbf{f}(\mathbf{x}, t), \mathbf{X}_i(t)\}. \quad (45)$$

Since the interpolation operator $\mathcal{I}_a[\mathbf{X}_i(t)]$ depends on the positions of the Lagrangian nodes, $\mathcal{P}$ does not commute with any of the operators $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{F}$. Therefore, the commonly used IBM algorithm

$$\{\mathbf{f}(\mathbf{x}, t+h), \mathbf{X}_i(t+h)\} = [\text{PSCF}] \{\mathbf{f}(\mathbf{x}, t), \mathbf{X}_i(t)\} \quad (46)$$

is only first-order accurate. It is straightforward to write down a formally second order scheme:

$$\{\mathbf{f}(\mathbf{x}, t+h), \mathbf{X}_i(t+h)\} = \left[\mathbf{P}^{\frac{1}{2}} \mathbf{F}^{\frac{1}{2}} (\mathbf{C}^{\frac{1}{2}} \mathbf{S} \mathbf{C}^{\frac{1}{2}}) \mathbf{F}^{\frac{1}{2}} \mathbf{P}^{\frac{1}{2}} \right] \{\mathbf{f}(\mathbf{x}, t), \mathbf{X}_i(t)\}. \quad (47)$$

During the position updates $\mathbf{P}^{\frac{1}{2}}$, the velocity is constant but the interpolation $\mathcal{I}_a[\mathbf{X}_i(t)]$ changes. Therefore a simple forward Euler update is insufficient, and the Crank-Nicolson rule becomes implicit. While one possibility is to use the midpoint rule in each of the two half-steps $\mathbf{P}^{\frac{1}{2}}$, we suggest here a simpler explicit scheme that uses a predictor step for the positions after the first half-step. Then we use a backward Euler in the first position half-step and a forward Euler in the second half step:

$$\begin{aligned} \tilde{\mathbf{X}}_i(t + \frac{h}{2}) &= \mathbf{X}_i(t) + \frac{h}{2} \mathcal{I}_a[\mathbf{X}_i(t)] \mathbf{u}(t), \\ \mathbf{X}_i(t + \frac{h}{2}) &= \mathbf{X}_i(t) + \frac{h}{2} \mathcal{I}_a[\tilde{\mathbf{X}}_i(t + \frac{h}{2})] \mathbf{u}(t), \\ \mathbf{X}_i(t+h) &= \mathbf{X}_i(t + \frac{h}{2}) + \frac{h}{2} \mathcal{I}_a[\tilde{\mathbf{X}}_i(t + \frac{h}{2})] \mathbf{u}(t+h). \end{aligned} \quad (48)$$

We get an overall position update

$$\mathbf{X}_i(t+h) = \mathbf{X}_i(t) + h\mathcal{I}_a[\tilde{\mathbf{X}}_i(t+\frac{h}{2})]\frac{\mathbf{u}(t) + \mathbf{u}(t+h)}{2}, \quad (49)$$

which is a trapezoidal quadrature of Eq. (43). This scheme can be seen as the adaption of the formally second-order method in Refs. [70, 71] to the lattice Boltzmann method. If the body force $\mathbf{F}(\mathbf{x}, t)$ depends on the positions $\mathbf{X}_i$ but not on the flow velocity $\mathbf{u}$, the body forces in the first and second half-steps $\mathbf{F}^{\frac{1}{2}}$ are the same. The application of the body forces thus relates to a midpoint scheme which is interleaved with the LB collisions and streaming.

$$\begin{aligned} \rho\mathbf{u}(t+h) &= \rho\mathbf{u}(t+\frac{h}{2}) + \frac{h}{2}\mathcal{I}_a^*[\tilde{\mathbf{X}}_i(t+\frac{h}{2})]\mathbf{F}_i(t+\frac{h}{2}) \\ &= \left[\mathbf{C}^{\frac{1}{2}}\mathbf{S}\mathbf{C}^{\frac{1}{2}}\right] \left(\rho\mathbf{u}(t) + \frac{h}{2}\mathcal{I}_a^*[\tilde{\mathbf{X}}_i(t+\frac{h}{2})]\mathbf{F}_i(t+\frac{h}{2})\right) \\ &\quad + \frac{h}{2}\mathcal{I}_a^*[\tilde{\mathbf{X}}_i(t+\frac{h}{2})]\mathbf{F}_i(t+\frac{h}{2}). \end{aligned} \quad (50)$$

The scheme in Eq. (47) is easily implemented by combining a Verlet-like update for the Lagrangian nodes $\mathbf{X}_i$ with any second order accurate LB scheme. A pseudo-code is listed in Algorithm 2.

5.2. Force coupling

The immersed boundary method works well if the dynamics of the fluid and the solute objects evolve on similar time scales. In soft matter systems, the internal motion of the solutes can be much faster than the fluid motion and hence needs to be resolved separately. This can be done by coupling LB to a molecular dynamics simulation where the immersed particles evolve according to Newton's equations of motion [13]. The particle positions $\mathbf{r}_i$ are now Lagrangian points that move with their own velocity $\mathbf{v}_i$, and the intrinsic interactions $\mathbf{F}_i$ enter Newton's equation of motion for the particles.

The fluid-particle interactions are incorporated by a coupling force that constrains the fluid flow to the motion of particles and vice-versa. This coupling force is modeled as a drag force proportional to the relative velocity of the particles with respect to the flow [13]

$$\mathbf{F}_i^h(t) = -\gamma[\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)] + \boldsymbol{\zeta}_i, \quad (51)$$

where γ is a friction constant, $\boldsymbol{\zeta}_i$ is a random force required to satisfy the fluctuation-dissipation theorem, and $\mathbf{u}(\mathbf{r}_i, t) = \mathcal{I}_a[\mathbf{r}_i(t)]\mathbf{u}(\mathbf{r}_i, t)$ is the fluid velocity at the particle's position. Since the coupling force must not change the total momentum of the combined fluid-particle system, an opposite force needs to be applied to the fluid in terms of body forces on the lattice sites within the interpolation range around the particle

$$\mathbf{F}^h(\mathbf{x}, t) = -\mathcal{I}_a^*[\mathbf{r}_i(t)]\mathbf{F}_i^h(t). \quad (52)$$

The interpolation and spreading operators are the same as in the immersed boundary method. The equations of motion for the particles are

$$\frac{d}{dt}\mathbf{r}_i(t) = \mathcal{P}\mathbf{r}_i(t) = \mathbf{v}_i(t), \quad (53a)$$

$$\frac{d}{dt}\mathbf{v}_i(t) = \mathcal{F}\mathbf{v}_i(t) = \frac{1}{m_i} \left(\mathbf{F}_i^h + \mathbf{F}_i^{\text{int}} \right), \quad (53b)$$

where $\mathbf{F}_i^{\text{int}}$ are intrinsic interactions between the particles.

The evolution of the coupled system is now given by

$$\frac{\partial}{\partial t} \{ \mathbf{f}(\mathbf{x}, t), \mathbf{r}_i(t), \mathbf{v}_i(t) \} = [\mathcal{S} + \mathcal{C} + \mathcal{F} + \mathcal{P}] \{ \mathbf{f}(\mathbf{x}, t), \mathbf{r}_i(t), \mathbf{v}_i(t) \}, \quad (54)$$

where the overall force operator $\mathcal{F}$ acts simultaneously on the particle velocities and on the fluid flow, cf. Eq. (27),

$$\mathcal{F}\mathbf{v}_i(t) = \mathbf{F}_i^h(t), \quad (55a)$$

$$\mathcal{F}\mathbf{u}(\mathbf{x}, t) = -\mathcal{I}_a^*[\mathbf{r}_i(t)]\mathbf{F}_i^h(t). \quad (55b)$$

As before, we can apply the Trotter-Suzuki decomposition formula to obtain a splitting scheme [28]

$$\{ \mathbf{f}, \mathbf{r}_i, \mathbf{v}_i \} (t + h) = \left[\mathbf{P}^{\frac{1}{2}} \mathbf{F}^{\frac{1}{2}} (\mathbf{C}^{\frac{1}{2}} \mathbf{S} \mathbf{C}^{\frac{1}{2}}) \mathbf{F}^{\frac{1}{2}} \mathbf{P}^{\frac{1}{2}} \right] \{ \mathbf{f}, \mathbf{r}_i, \mathbf{v}_i \} (t). \quad (56)$$

This is basically a Verlet algorithm with a lattice Boltzmann update in the middle of the velocity update. Since the hydrodynamic coupling forces are dependent on both the particle velocities and the fluid flow, the equations of motion are coupled

$$\frac{\partial}{\partial t} \mathbf{v}_i(t) = -\frac{1}{m_i} [\gamma (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) - \boldsymbol{\zeta}_i - \mathbf{F}_i^{\text{int}}], \quad (57a)$$

$$\frac{\partial}{\partial t} \mathbf{u}(\mathbf{r}_i, t) = \frac{1}{\rho a^3} [\gamma (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) - \boldsymbol{\zeta}_i]. \quad (57b)$$

An accurate midpoint scheme for the velocities and the fluid flow is

$$\mathbf{v}_i(t + h) = \mathbf{v}_i(t) + \frac{h}{m_i} \mathbf{F}_i^h(t + \frac{h}{2}) + \frac{h}{m_i} \mathbf{F}_i^{\text{int}}, \quad (58a)$$

$$\mathbf{u}(\mathbf{r}_i, t + h) = \mathbf{u}(\mathbf{r}_i, t) - \frac{h}{\rho} \mathbf{F}_i^h(t + \frac{h}{2}). \quad (58b)$$

In Ref. [15], a constant flow velocity $\mathbf{u}(\mathbf{r}_i, t)$ is assumed to determine the viscous coupling force. This is justified for slowly varying flow fields that are only slightly perturbed by the coupling, but strictly speaking it does not give an $O(h^2)$ approximation of Eq. (57). Therefore, we directly apply Crank-Nicolson

discretization here

$$\frac{\mathbf{v}_i(t+h) - \mathbf{v}_i(t)}{h} = -\frac{\gamma}{m_i} \frac{\mathbf{v}_i(t+h) - \mathbf{u}(\mathbf{r}_i, t+h)}{2} \quad (59a)$$

$$\begin{aligned} & -\frac{\gamma}{m_i} \frac{\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)}{2} + \frac{1}{m_i} (\mathbf{F}_i^{\text{int}} + \boldsymbol{\zeta}_i), \\ \frac{\mathbf{u}(\mathbf{r}_i, t+h) - \mathbf{u}(\mathbf{r}_i, t)}{h} &= \frac{\gamma}{\rho a^3} \frac{\mathbf{v}_i(t+h) - \mathbf{u}(\mathbf{r}_i, t+h)}{2} \quad (59b) \\ & + \frac{\gamma}{\rho a^3} \frac{\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)}{2} - \frac{1}{\rho a^3} \boldsymbol{\zeta}_i, \end{aligned}$$

and obtain

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) - \frac{\gamma^{-1}\alpha}{1 + \frac{\alpha}{2} + \frac{\beta}{2}} \left[\gamma (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) - \boldsymbol{\zeta}_i - \frac{2+\beta}{2} \mathbf{F}_i^{\text{int}} \right], \quad (60)$$

$$\mathbf{u}(\mathbf{r}_i, t+h) = \mathbf{u}(\mathbf{r}_i, t) + \frac{\gamma^{-1}\beta}{1 + \frac{\alpha}{2} + \frac{\beta}{2}} \left[\gamma (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) - \boldsymbol{\zeta}_i + \frac{\alpha}{2} \mathbf{F}_i^{\text{int}} \right]. \quad (61)$$

where the parameters $\alpha = \frac{\gamma h}{m_i}$ and $\beta = \frac{\gamma h}{\rho a^3}$ were introduced. Requiring consistency between (58) and (59) we obtain the midpoint force

$$\mathbf{F}_i^h(t + \frac{h}{2}) = -\frac{1}{1 + \frac{\alpha}{2} + \frac{\beta}{2}} \left[\gamma (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) - \boldsymbol{\zeta}_i + \frac{\alpha}{2} \mathbf{F}_i^{\text{int}} \right]. \quad (62)$$

The denominator $1 + \frac{\alpha}{2} + \frac{\beta}{2}$ has the typical form of a discrete correction, cf. Eq. (19), and takes into account that both $\mathbf{v}_i(t)$ and $\mathbf{u}(\mathbf{r}_i, t)$ change during the time step. Eq. (62) is straightforward to implement and can lead to more accurate integration of the coupling forces [15]. Note that the formulas (58)-(62) refer to the full time step operator $\mathbf{F}$, and a half-step $\mathbf{F}^{\frac{1}{2}}$ is achieved by substituting h by $\frac{h}{2}$. For a full time step in the splitting scheme (56), the procedure is applied before and after the LB update. A pseudo-code for force coupling is listed in Algorithm 3.

It is worthwhile to note that the viscous force coupling does not lead to an instantaneous stick boundary condition. To see this, we calculate the difference between the particle velocity and fluid flow after the force application

$$\begin{aligned} \mathbf{v}_i(t+h) - \mathbf{u}(\mathbf{r}_i, t+h) &= \frac{1 - \frac{\alpha}{2} - \frac{\beta}{2}}{1 + \frac{\alpha}{2} + \frac{\beta}{2}} (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) \\ &+ \frac{\alpha + \beta}{1 + \frac{\alpha}{2} + \frac{\beta}{2}} \frac{\boldsymbol{\zeta}_i}{\gamma} + \frac{\alpha}{1 + \frac{\alpha}{2} + \frac{\beta}{2}} \frac{\mathbf{F}_i^{\text{int}}}{\gamma}. \quad (63) \end{aligned}$$

The second term stems from the thermal fluctuations and is zero on average. Requiring that the coefficient of the velocity difference vanishes yields $\alpha + \beta = 2$ and an expression for the friction

$$\gamma = \frac{\rho a^3}{h} \frac{2\beta}{\alpha + \beta} = \frac{\rho a^3}{h} \frac{2}{1 + \frac{\rho a^3}{m_i}}. \quad (64)$$

In this case, the velocity difference after force application is

$$\mathbf{v}_i(t+h) - \mathbf{u}(\mathbf{r}_i, t+h) = \frac{h}{2m_i} \mathbf{F}_i^{\text{int}} + \frac{\boldsymbol{\zeta}_i}{\gamma}, \quad (65)$$

which does not vanish on average. The reason for this spurious velocity slip lies in the nature of the discretization: the velocities are effectively evaluated at the midpoint and therefore the intrinsic forces acting during the second half-interval are not accounted for. This does not cause a problem if one considers point-like particles that are dragged through the fluid. In this case, one can tune γ to obtain a defined physical friction and an effective particle size. The effective physical friction has to be re-normalized due to the spatial interpolation on the discrete lattice and leads to an effective size R_{eff} of the particles [9, 13]

$$\frac{1}{R_{\text{eff}}} = \frac{6\pi\eta}{\gamma} + \frac{1}{ga}. \quad (66)$$

Moreover, since the model is based on point particles, extended objects will necessarily be under-resolved when represented by a single particle. Therefore it is necessary to add fluctuations $\boldsymbol{\zeta}_i$ to the coupling force in order to satisfy the fluctuation dissipation relation [9, 13].

In the case of more complicated objects like cells and vesicles with specific shapes, it may be more appropriate to represent the object by a set of surface points that are coupled to the fluid. The spurious slip velocity (65) may then be undesirable. A remedy for this situation is a modification of the external boundary force as described in the following section.

5.3. External boundary force (EBF)

The modeling of the fluid-particle interactions as a viscous force has proved successful for simulating polymer chains in solution, where the individual beads of the chain experience the viscous coupling [13–15], but it may be less appropriate for extended objects. In this case, the fluid velocity at the surface is typically required to satisfy a no-slip boundary condition, i.e., the fluid and the object should move with the same velocity. Therefore, the fluid-particle interaction can also be modeled as an external boundary force that forces the fluid to move with the surface velocity [18, 72]. In the latter references, the interaction force acting at a surface point is

$$\mathbf{F}_i^h(\mathbf{r}_i, t) = -\frac{\rho a^3}{h} (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)), \quad (67)$$

which is equivalent to the force coupling (51) with $\gamma = \frac{\rho a^3}{h}$. In this section we skip the stochastic force $\boldsymbol{\zeta}_i$ which is straightforward to include if demanded by the fluctuation dissipation relation for the immersed object.

As we have seen above, however, the force in (67) can not lead to the no-slip boundary condition if intrinsic forces are present. The reason is that, in contrast to the immersed boundary method, the intrinsic forces are applied to

the particles and accelerate them relative to the fluid. In order to enforce the no-slip boundary condition even in the presence of intrinsic forces, we require

$$\begin{aligned} \mathbf{v}_i(t+h) &= \mathbf{u}(\mathbf{r}_i, t+h), \\ \mathbf{v}_i(t) + \frac{h}{m_i} \left[\mathbf{F}_i^h(t + \frac{h}{2}) + \mathbf{F}_i^{\text{int}} \right] &= \mathbf{u}(\mathbf{r}_i, t) - \frac{h}{\rho a^3} \mathbf{F}_i^h(t + \frac{h}{2}). \end{aligned} \quad (68)$$

Solving for the fluid-particle interaction force we get

$$\mathbf{F}_i^h(t + \frac{h}{2}) = -\frac{\gamma}{\alpha + \beta} (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) - \frac{\alpha}{\alpha + \beta} \mathbf{F}_i^{\text{int}}. \quad (69)$$

The updates for the particle and flow velocity are thus

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) - \frac{\alpha}{\alpha + \beta} (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) + \frac{h}{m_i} \frac{\beta}{\alpha + \beta} \mathbf{F}_i^{\text{int}}, \quad (70a)$$

$$\mathbf{u}(\mathbf{r}_i, t) = \mathbf{u}(\mathbf{r}_i, t) + \frac{\beta}{\alpha + \beta} (\mathbf{v}_i(t) - \mathbf{u}(\mathbf{r}_i, t)) + \frac{h}{\rho a^3} \frac{\alpha}{\alpha + \beta} \mathbf{F}_i^{\text{int}}. \quad (70b)$$

The modified EBF in Eq. (70) leads to an instantaneous no-slip boundary condition at the fluid-object interface. The implementation is straightforward and has the same structure as the force coupling in Algorithm 3.

If we introduce

$$r = \frac{\alpha}{\alpha + \beta} = \frac{1}{1 + \frac{m_i}{\rho a^3}}, \quad (1-r) = \frac{\beta}{\alpha + \beta} = \frac{1}{1 + \frac{\rho a^3}{m_i}}, \quad (71)$$

Equations (70) can be interpreted as follows: A fraction r of the intrinsic force is included in the hydrodynamic coupling and thus applied to the fluid, while the remaining fraction $1-r$ is applied to the particle system. We remark that the same equations are obtained, if we start out by assuming that the intrinsic force is split between the fluid and the particles. That is, Eqs. (70) are a second order accurate scheme for the differential equations

$$\frac{\partial}{\partial t} \mathbf{v}_i(t) = -\frac{1}{m_i} [\gamma (\mathbf{v}_i - \mathbf{u}(\mathbf{r}_i, t)) - \boldsymbol{\zeta}_i - (1-r) \mathbf{F}_i^{\text{int}}], \quad (72a)$$

$$\frac{\partial}{\partial t} \mathbf{u}(\mathbf{r}_i, t) = \frac{1}{\rho a^3} [\gamma (\mathbf{v}_i - \mathbf{u}(\mathbf{r}_i, t)) - \boldsymbol{\zeta}_i + r \mathbf{F}_i^{\text{int}}]. \quad (72b)$$

Equations (64) and (71) are then the conditions under which the no-slip boundary condition is satisfied.

It is interesting to note that the distribution of the intrinsic force is controlled by the ratio between the particle mass m_i and the fluid mass within a unit cell ρa^3 , where a is the lattice spacing. In the limit $m_i \gg \rho a^3$ we have $r \rightarrow 0$, and the original EBF is recovered, where the intrinsic force is applied to the particles only. Conversely, in the limit $m_i \ll \rho a^3$ we have $r \rightarrow 1$ and recover the IBM where the intrinsic force is applied to the fluid and the particles are simply advected with the flow velocity. This shows that the immersed boundary method

corresponds to the limit where the particle mass becomes negligible compared to the fluid mass. The dimensionless parameter r controls whether the fluid is constrained to the particle velocity, or whether the particles are constrained to the fluid flow. We therefore refer to r as the “immersion number”. It measures the relative importance of inertia of the particles and the fluid. The immersion number also depends on resolution: For fixed ρ and m_i , r goes to zero in the continuum limit $a \rightarrow 0$. For fixed resolution, r depends only on the fraction $\alpha/\beta = \rho a^3/m_i$ and is independent of γ . Hence r can be tuned to control the strength of the coupling with respect to the transfer of intrinsic forces to the fluid. It is often desirable to have a neutrally buoyant object, which is achieved when the immersion number is $r = 1/2$ or equivalently $m_i = \rho a^3$. This choice is also typical in other mesoscopic methods, for example, in coupling elastic cells to multi-particle collision dynamics [73].

The payoff for satisfying the no-slip boundary condition instantaneously is the loss of freedom in tuning γ . The physical friction and effective size of individual point-like particles can not be tuned as easily any more. However, this is fully compatible with the representation of the solute object by a set of surface points: The object representation has an explicit size, where the Lagrangian points are rather markers than bead-like particles. The effective physical parameters of the object are thus primarily determined by the boundary conditions on the surface.

6. Multiple time-step integration of force coupled systems

Since the intrinsic forces are typically short-range, they often require a time-step Δt much smaller than a typical LB step h . Reducing the overall time step can be costly, and as the dominant computational effort is the fluid update, it is desirable to maintain a large fluid time-step h . For accurate integration of the intrinsic forces, we can sub-divide h into an even integer number $2n$ of sub-steps of size $\Delta t = h/(2n)$. Such a multiple time step scheme can be written in terms of operator splitting as follows [23]

$$\begin{aligned} e^{h(S+C+\mathcal{P}+\mathcal{F})} &\approx \left[e^{n\Delta t(\mathcal{P}+\mathcal{F})} \right] e^{h(S+C)} \left[e^{n\Delta t(\mathcal{P}+\mathcal{F})} \right] \\ &= \left[e^{\Delta t(\mathcal{P}+\mathcal{F})} \right]^n e^{h(S+C)} \left[e^{\Delta t(\mathcal{P}+\mathcal{F})} \right]^n \\ &\approx \left[P^{\frac{1}{4n}} F^{\frac{1}{2n}} P^{\frac{1}{4n}} \right]^n [C^{\frac{1}{2}} S C^{\frac{1}{2}}] \left[P^{\frac{1}{4n}} F^{\frac{1}{2n}} P^{\frac{1}{4n}} \right]^n. \end{aligned} \quad (73)$$

Equation (73) are two Verlet-like updates, each iterated n times with a time step Δt , and arranged symmetrically around the LB update. The symmetric arrangement is essential for accurate operator splitting, but it seems to receive little attention in common software implementations.

Finally, we note that it is also possible to split the update of the full system in the following way

$$e^{h(S+C+\mathcal{P}+\mathcal{F})} \approx F_h^{\frac{1}{2}} \left[P^{\frac{1}{4n}} F_{\text{int}}^{\frac{1}{2n}} P^{\frac{1}{4n}} \right]^n [C^{\frac{1}{2}} S C^{\frac{1}{2}}] \left[P^{\frac{1}{4n}} F_{\text{int}}^{\frac{1}{2n}} P^{\frac{1}{4n}} \right]^n F_h^{\frac{1}{2}}, \quad (74)$$

where we have decomposed the force operator into the intrinsic part F_{int} and the coupling part F_h . In this arrangement, the particle system can be integrated using an ODE solver such as LSODE or VODE [74, 75]. The fluid-particle interactions are arranged at the outside in order to satisfy the no-slip boundary condition after a full cycle through all parts. Moreover, the number of evaluations of the hydrodynamic coupling force is reduced by a factor n . The payoff is that this may lead to inaccuracies in tracing the flow field by the particle positions. A detailed numerical analysis of the accuracy, efficiency and stability of the above splitting schemes is left for future work.

7. Conclusions

The operator splitting approach for the lattice Boltzmann method makes it straightforward to derive accurate time-marching schemes for various coupling methods. The proposed second-order schemes for the immersed boundary method, force coupling and external boundary force improve the accuracy compared to commonly implemented first-order algorithms. The modified external boundary force unveils the close relation of the immersed boundary method and force coupling. All three methods can be cast into the form of force coupling and thus have a unified foundation. The coupling strength can be quantified by the ratio between fluid and particle mass. In practice, this allows us to control the coupling by appropriate choice of the immersion number.

While finite-time accuracy of the solutions may not be the main concern if statistical properties in equilibrium or non-equilibrium steady states are evaluated, accurate integration methods are still favorable to maintain long-time stability [34, 35, 76]. Splitting methods are fairly established in molecular dynamics simulations and lead to superior algorithms, in particular when applied to systems with multiple scales and long-range forces [23, 29, 30]. It is therefore anticipated that the fluid-particle coupling methods discussed in this work will also benefit from a systematic operator splitting approach.

Acknowledgments

I would like to thank Marisol Ripoll, Roland G. Winkler, Burkhard Dünweg and Anthony J. C. Ladd for valuable discussions and helpful comments. I also appreciate the careful and constructive review by an anonymous referee. Financial support from Volkswagen Stiftung under Grant No. I/83576 is gratefully acknowledged.

Appendix A. Spatial interpolation functions

In the following, we outline the construction of the spatial interpolation functions which are used in the hybrid coupling schemes to map between the Eulerian grid and the Lagrangian particle system. More details can be found, e.g., in [20].

The three-dimensional delta function $\delta_a(\mathbf{x})$ is taken as a product of one-dimensional functions

$$\delta_a(\mathbf{x}) = \phi\left(\frac{x}{a}\right)\phi\left(\frac{y}{a}\right)\phi\left(\frac{z}{a}\right), \quad (\text{A.1})$$

where x, y, z are the components of the vector $\mathbf{x}$. By postulate, the function $\phi(r)$ should be continuous and satisfy the following conditions [20]

$$\sum_j \phi(r-j) = 1 \quad \forall r, \quad (\text{A.2a})$$

$$\sum_j j\phi(r-j) = r \quad \forall r, \quad (\text{A.2b})$$

$$\sum_j (\phi(r-j))^2 = C \quad \forall r, \quad (\text{A.2c})$$

where the sum runs over all integer numbers j .

For reasons of computational efficiency, $\phi(r)$ should also have bounded support. Conditions (A.2a) and (A.2b) imply

$$\sum_{\mathbf{x}} \delta_a(\mathbf{x} - \mathbf{X}) = 1, \quad (\text{A.3a})$$

$$\sum_{\mathbf{x}} \mathbf{x} \delta_a(\mathbf{x} - \mathbf{X}) = \mathbf{X}, \quad (\text{A.3b})$$

where the sum over $\mathbf{x}$ runs over all nodes of the Eulerian grid, i.e., $\mathbf{x} = (ka, la, ma)$ and k, l, m run over all integer numbers. These conditions guarantee that the total force and torque on the Eulerian grid are the same as on the Lagrangian grid

$$\sum_{\mathbf{x}} a^3 \mathbf{F}(\mathbf{x}, t) = \sum_{\mathbf{x}} \sum_i \mathbf{F}_i(t) \delta_a(\mathbf{x} - \mathbf{X}_i) = \sum_i \mathbf{F}_i(t), \quad (\text{A.4a})$$

$$\sum_{\mathbf{x}} \mathbf{x} \times a^3 \mathbf{F}(\mathbf{x}, t) = \sum_{\mathbf{x}} \sum_i \mathbf{x} \times \mathbf{F}_i(t) \delta_a(\mathbf{x} - \mathbf{X}_i) = \sum_i \mathbf{X}_i \times \mathbf{F}_i(t). \quad (\text{A.4b})$$

Moreover, (A.2a) and (A.2b) imply that smooth functions are interpolated with second-order spatial accuracy. The third condition (A.2c) can also be written as a Schwarz inequality [20]

$$\left| \sum_j \phi(r_1 - j) \phi(r_2 - j) \right| \leq C, \quad (\text{A.5})$$

where the equal sign holds for $r_1 = r_2$. This ensures that the overlap of two Lagrangian particles is strongest when their positions coincide, and the strength is independent of position. Condition (A.2c) is thus a substitute for Galilean invariance, which is impossible to satisfy if $\phi(r)$ has bounded support, cf. [20].

The three conditions (A.2a)-(A.2b) can be used to determine a three-point interpolation function $\phi_3(r)$

$$\phi_3(r) = \begin{cases} \frac{1}{3} (1 + \sqrt{1 - 3r^2}) & 0 \leq |r| \leq \frac{1}{2} \\ \frac{1}{6} (5 - 3|r| - \sqrt{6|r| - 2 - 3r^2}) & \frac{1}{2} \leq |r| \leq \frac{3}{2} \\ 0 & \frac{3}{2} \leq |r|. \end{cases} \quad (\text{A.6})$$

This function has a continuous derivative $\phi'_3(r)$, in contrast to the linear interpolation used, e.g. in [13]. It therefore has the advantage that the flow gradient $\nabla \mathbf{u}$ varies smoothly and the particles do not experience jumps when crossing grid lines. A numerical analysis of the properties of different interpolation functions can be found in [9].

Algorithm 1 Collide-Stream-Collide Lattice Boltzmann

Ensure: Parallelization has communicated processor boundaries

```
procedure LBUPDATE
  for all  $(x, y, z)$  do
    LBPUSH( $x, y, z$ )
    if  $(x, y, z) \in Interior$  then LBREAD( $x - 1, y - 1, z - 1$ )
    end if
  end for
   $t \leftarrow t + h$ 
end procedure
```

Require: Random numbers $r_k(x, y, z)$

```
procedure LBPUSH( $x, y, z$ )
  for all  $k$  do
     $m_k(x, y, z) \leftarrow m_k^{eq}(x, y, z) + \gamma_k m_k^{neq}(x, y, z) + \sqrt{\rho} \varphi_k r_k(x, y, z)$ 
     $f_i(x + c_{ix}, y + c_{iy}, z + c_{iz}) \leftarrow \sum_k a^{c_i}/b_k \cdot e_{ki} m_k(x, y, z) + G_i$ 
  end for
end procedure
```

Require: Random numbers $r_k(x, y, z)$

```
procedure LBREAD( $x, y, z$ )
  for all  $k$  do
     $m_k(x, y, z) \leftarrow \sum_i e_{ki} f_i(x, y, z)$ 
     $m_k(x, y, z) \leftarrow m_k^{eq}(x, y, z) + \gamma_k m_k^{neq}(x, y, z) + \sqrt{\rho} \varphi_k r_k(x, y, z) + \sum_i e_{ki} G_i$ 
  end for
end procedure
```

Algorithm 2 Immersed Boundary Method

```
procedure IMMERSEDBOUNDARYMETHOD
  for all  $i \in \text{Particles}$  do
     $\tilde{\mathbf{X}}_i \leftarrow \mathbf{X}_i + h/2 \cdot \mathcal{I}_a[\mathbf{X}_i]\mathbf{u}$ 
  end for
   $\mathbf{F}_i \leftarrow \text{CALCULATEFORCES}(\tilde{\mathbf{X}}_i)$ 
  for all  $i \in \text{Particles}$  do
     $\mathbf{X}_i \leftarrow \mathbf{X}_i + h/2 \cdot \mathcal{I}_a[\tilde{\mathbf{X}}_i]\mathbf{u}$ 
     $\text{SPREADFORCE}(\mathcal{I}_a^*[\tilde{\mathbf{X}}_i]\mathbf{F}_i)$ 
  end for
  LBUPDATE
  for all  $i \in \text{Particles}$  do
     $\text{SPREADFORCE}(\mathcal{I}_a^*[\tilde{\mathbf{X}}_i]\mathbf{F}_i)$ 
     $\mathbf{X}_i \leftarrow \mathbf{X}_i + h/2 \cdot (\mathbf{V}_i + \mathcal{I}_a[\tilde{\mathbf{X}}_i]\mathbf{u})$ 
  end for
end procedure
```

Algorithm 3 Force Coupling

Require: Random forces ζ_i

```
procedure CALCULATECOUPLING( $\mathbf{r}_i, \mathbf{v}_i, \mathbf{F}_i, \mathbf{u}$ )
  for all  $i \in \text{Particles}$  do
     $\mathbf{F}_i^h \leftarrow -1/(1 + \alpha/2 + \beta/2) \cdot (\gamma(\mathbf{v}_i - \mathcal{I}_a[\mathbf{r}_i]\mathbf{u}) - \zeta_i + \alpha/2 \cdot \mathbf{F}_i)$ 
  end for
end procedure
```

```
procedure FORCECOUPLING
  for all  $i \in \text{Particles}$  do
     $\mathbf{r}_i \leftarrow \mathbf{r}_i + \Delta t/2 \cdot \mathbf{v}_i$ 
  end for
   $\mathbf{F}_i \leftarrow \text{CALCULATEFORCES}(\mathbf{r}_i)$ 
  for all  $i \in \text{Particles}$  do
     $\mathbf{F}_i^h \leftarrow \text{CALCULATECOUPLING}(\mathbf{r}_i, \mathbf{v}_i, \mathbf{F}_i, \mathbf{u})$ 
     $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t/2 \cdot (\mathbf{F}_i^h + \mathbf{F}_i)$ 
     $\text{SPREADFORCE}(-\mathcal{I}_a^*[\mathbf{r}_i]\mathbf{F}_i^h)$ 
  end for
  LBUPDATE
  for all  $i \in \text{Particles}$  do
     $\mathbf{F}_i^h \leftarrow \text{CALCULATECOUPLING}(\mathbf{r}_i, \mathbf{v}_i, \mathbf{F}_i, \mathbf{u})$ 
     $\text{SPREADFORCE}(-\mathcal{I}_a^*[\mathbf{r}_i]\mathbf{F}_i^h)$ 
     $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t/2 \cdot (\mathbf{F}_i^h + \mathbf{F}_i)$ 
     $\mathbf{r}_i \leftarrow \mathbf{r}_i + \Delta t/2 \cdot \mathbf{v}_i$ 
  end for
end procedure
```

References

- [1] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*, Clarendon Press, 2001.
- [2] X. He, L.-S. Luo, A priori derivation of the lattice Boltzmann equation, *Phys. Rev. E* 55 (1997) R6333–R6336.
- [3] X. He, L.-S. Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *Phys. Rev. E* 56 (1997) 6811–6817.
- [4] X. Shan, X. He, Discretization of the velocity space in the solution of the Boltzmann equation, *Phys. Rev. Lett.* 80 (1998) 65–68.
- [5] M. Junk, A. Klar, L. S. Luo, Asymptotic analysis of the lattice Boltzmann equation, *J. Comp. Phys.* 210 (2005) 676–704.
- [6] M. Krafczyk, J. Tölke, E. Rank, M. Schulz, Two-dimensional simulation of fluid-structure interaction using lattice-Boltzmann methods, *Computers & Structures* 79 (2001) 2031–2037.
- [7] S. Geller, J. Tölke, M. Krafczyk, Lattice-Boltzmann method on quadtree-type grids for fluid-structure interaction, in: H.-J. Bungartz, M. Schäfer (Eds.), *Lecture Notes in Computational Science and Engineering*, volume 53, Springer, 2006, pp. 270–293.
- [8] A. De Rosis, G. Falcucci, S. Ubertini, F. Ubertini, A coupled lattice Boltzmann-finite element approach for two-dimensional fluid-structure interaction, *Computers & Fluids* 86 (2013) 558–568.
- [9] B. Dünweg, A. J. C. Ladd, Lattice Boltzmann simulations of soft matter systems, *Adv. Poly. Sci.* 221 (2008) 89.
- [10] C. K. Aidun, J. R. Clausen, Lattice-Boltzmann method for complex flows, *Annu. Rev. Fluid Mech.* 42 (2010) 439–472.
- [11] A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation, *J. Fluid Mech.* 271 (1994) 285–309.
- [12] A. J. C. Ladd, R. Verberg, Lattice-Boltzmann simulations of particle-fluid suspensions, *J. Stat. Phys.* 104 (2001) 1191–1251.
- [13] P. Ahlrichs, B. Dünweg, Simulation of a single polymer chain in solution by combining lattice Boltzmann and molecular dynamics, *J. Chem. Phys.* 111 (1999) 8225–8239.
- [14] T. T. Pham, U. D. Schiller, J. R. Prakash, B. Duenweg, Implicit and explicit solvent models for the simulation of a single polymer chain in solution: Lattice Boltzmann vs Brownian dynamics, *J. Chem. Phys.* 131 (2009) 164114.

- [15] A. J. C. Ladd, R. Kekre, J. E. Butler, Comparison of the static and dynamic properties of a semiflexible polymer using lattice Boltzmann and Brownian-dynamics simulations, *Phys. Rev. E* 80 (2009) 036704.
- [16] A. Alexeev, R. Verberg, A. C. Balazs, Modeling the interactions between deformable capsules rolling on a compliant surface, *Soft Matter* 2 (2006) 499–509.
- [17] M. M. Dupin, I. Halliday, C. M. Care, A. Lyuba, L. L. Munn, Modeling the flow of dense suspensions of deformable particles in three dimensions, *Phys. Rev. E* 75 (2007) 066707.
- [18] J. Wu, C. K. Aidun, Simulating 3D deformable particle suspensions using lattice Boltzmann method with discrete external boundary force, *Int. J. Numer. Meth. Fluids* 62 (2010) 765–783.
- [19] R. A. Brownlee, A. N. Gorban, J. Levesley, Stability and stabilization of the lattice Boltzmann method, *Phys. Rev. E* 75 (2007) 036711.
- [20] C. S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [21] M. Serrano, G. De Fabritiis, P. Español, P. Coveney, A stochastic Trotter integration scheme for dissipative particle dynamics, *Math. Comp. Sim.* 72 (2006) 190–194.
- [22] T. Schlick, M. Mandziuk, R. D. Skeel, K. Srinivas, Nonlinear resonance artifacts in molecular dynamics simulations, *J. Comp. Phys.* 140 (1998) 1–29.
- [23] M. Tuckerman, B. J. Berne, G. J. Martyna, Reversible multiple time scale molecular dynamics, *J. Chem. Phys.* 97 (1992) 1990–2001.
- [24] A. Ricci, G. Ciccotti, Algorithms for Brownian dynamics, *Mol. Phys.* 101 (2003) 1927–1931.
- [25] G. Bussi, M. Parrinello, Accurate sampling using Langevin dynamics, *Phys. Rev. E* 75 (2007) 056707.
- [26] H. F. Trotter, On the product of semi-groups of operators, *Proc. Amer. Math. Soc.* 10 (1959) 545–551.
- [27] T. Kato, Trotter’s product formula for an arbitrary pair of self-adjoint contraction semigroups, in: *Topics in functional analysis (essays dedicated to M. G. Krein on the occasion of his 70th birthday)*, volume 3 of *Adv. in Math. Suppl. Stud.*, Academic Press, New York, 1978, pp. 185–195.
- [28] M. Suzuki, Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics, *J. Math. Phys.* 26 (1985) 601–612.

- [29] R. I. McLachlan, G. R. W. Quispel, Splitting methods, *Acta Numer.* 11 (2002) 341–434.
- [30] N. Bou-Rabee, Time integrators for molecular dynamics, *Entropy* 16 (2013) 138–162.
- [31] P. J. Dellar, An interpretation and derivation of the lattice Boltzmann method using Strang splitting, *Comput. Math. Appl.* 65 (2013) 129–141.
- [32] D. Frenkel, B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Academic Press, 2001.
- [33] R. Skeel, G. Zhang, T. Schlick, A family of symplectic integrators: Stability, accuracy, and molecular dynamics applications, *SIAM J. Sci. Comput.* 18 (1997) 203–222.
- [34] T. Shardlow, Splitting for dissipative particle dynamics, *SIAM J. Sci. Comput.* 24 (2003) 1267–1282.
- [35] G. De Fabritiis, M. Serrano, P. Español, P. Coveney, Efficient numerical integrators for stochastic models, *Physica A* 361 (2006) 429–440.
- [36] A. Malevanets, R. Kapral, Mesoscopic model for solvent dynamics, *J. Chem. Phys.* 110 (1999) 8605–8613.
- [37] G. Gompper, T. Ihle, D. M. Kroll, R. G. Winkler, Multi-particle collision dynamics: A particle-based mesoscale simulation approach to the hydrodynamics of complex fluids, *Adv. Poly. Sci.* 221 (2009) 1–87.
- [38] C. Cercignani, *The Boltzmann Equation and Its Applications*, volume 67 of *Applied Mathematical Sciences*, Springer, 1988.
- [39] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems, *Phys. Rev.* 94 (1954) 511–525.
- [40] E. P. Gross, E. A. Jackson, Kinetic models and the linearized Boltzmann equation, *Phys. Fluids* 2 (1959) 432–441.
- [41] X. Shan, X.-F. Yuan, H. Chen, Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation, *J. Fluid Mech.* 550 (2006) 413–441.
- [42] J. E. Broadwell, Study of rarefied shear flow by the discrete velocity method, *J. Fluid Mech.* 19 (1964) 401–414.
- [43] L.-S. Luo, Some recent results on discrete velocity models and ramifications for lattice Boltzmann equation, *Comp. Phys. Comm.* 129 (2000) 63–74.
- [44] X. He, X. Shan, G. D. Doolen, Discrete Boltzmann equation model for nonideal gases, *Phys. Rev. E* 57 (1998) R13–R16.

- [45] R. Benzi, S. Succi, M. Vergalossa, The lattice Boltzmann equation: theory and applications, *Phys. Rep.* 222 (1992) 145–197.
- [46] S. Chen, G. D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* 30 (1998) 329–364.
- [47] X. He, S. Chen, G. D. Doolen, A novel thermal model for the lattice Boltzmann method in incompressible limit, *J. Comp. Phys.* 146 (1998) 282–300.
- [48] Z. Guo, C. Zheng, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, *Phys. Rev. E* 65 (2002) 046308.
- [49] P. J. Dellar, Incompressible limits of lattice Boltzmann equations using multiple relaxation times, *J. Comp. Phys.* 190 (2003) 351–370.
- [50] M. Hénon, Viscosity of a lattice gas, *Complex Sys.* 1 (1987) 763–789.
- [51] K. Xu, X. He, Lattice Boltzmann method and gas-kinetic BGK scheme in the low-Mach number viscous flow simulations, *J. Comp. Phys.* 190 (2003) 100–117.
- [52] T. Ohwada, Higher order approximation methods for the Boltzmann equation, *J. Comp. Phys.* 139 (1998) 1–14.
- [53] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (1968) 506–517.
- [54] W. Gröbner, *Die Lie-Reihen und ihre Anwendungen*, Deutscher Verlag der Wissenschaften, 1960.
- [55] V. S. Varadarajan, *Lie groups, Lie algebras, and their representations*, Prentice-Hall, 1974.
- [56] E. Hairer, C. Lubich, G. Wanner, *Geometric Numerical Integration*, 2nd ed., Springer, 2006, pp. 51–96.
- [57] H. Yoshida, Construction of higher order symplectic integrators, *Physics Letters A* 150 (1990) 262 – 268.
- [58] A. Bobylev, T. Ohwada, The error of the splitting scheme for solving evolutionary equations, *Appl. Math. Lett.* 14 (2001) 45–48.
- [59] R. Salmon, The lattice Boltzmann method as a basis for ocean circulation modeling, *J. Marine Res.* 57 (1999) 503–535.
- [60] R. Salmon, Lattice Boltzmann solutions of the three-dimensional planetary geostrophic equations, *J. Marine Res.* 57 (1999) 847–884.
- [61] R. Adhikari, K. Stratford, M. E. Cates, A. J. Wagner, Fluctuating lattice Boltzmann, *Europhys. Lett.* 71 (2005) 473.

- [62] B. Dünweg, U. D. Schiller, A. J. C. Ladd, Statistical mechanics of the fluctuating lattice Boltzmann equation, *Phys. Rev. E* 76 (2007) 036704.
- [63] L.-S. Luo, Unified theory of lattice Boltzmann models for nonideal gases, *Phys. Rev. Lett.* 81 (1998) 1618–1621.
- [64] N. S. Martys, X. Shan, H. Chen, Evaluation of the external force term in the discrete Boltzmann equation, *Phys. Rev. E* 58 (1998) 6855–6857.
- [65] S. Ubertini, P. Asinari, S. Succi, Three ways to lattice Boltzmann: A unified time-marching picture, *Phys. Rev. E* 81 (2010) 016311.
- [66] S. Chapman, T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases*, Cambridge Mathematical Library Series, 3rd ed., Cambridge University Press, 1990.
- [67] G. McNamara, B. Alder, Analysis of the lattice Boltzmann treatment of hydrodynamics, *Physica A* 194 (1993) 218–228.
- [68] P. J. Dellar, Bulk and shear viscosities in lattice Boltzmann equations, *Phys. Rev. E* 64 (2001) 031203.
- [69] M. Wittmann, T. Zeiser, G. Hager, G. Wellein, Comparison of different propagation steps for lattice Boltzmann methods, *Comput. Math. Appl.* 65 (2013) 924–935.
- [70] M.-C. Lai, C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comp. Phys.* 160 (2000) 705–719.
- [71] Y. Mori, C. S. Peskin, Implicit second-order immersed boundary methods with boundary mass, *Comput. Meth. Appl. Mech. Eng.* 197 (2008) 2049–2067.
- [72] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comp. Phys.* 105 (1993) 354–366.
- [73] M. Peltomäki, G. Gompper, Sedimentation of single red blood cells, *Soft Matter* 9 (2013) 8346–8358.
- [74] A. C. Hindmarsh, ODEPACK, a systematized collection of ODE solvers, *IMACS Trans. Sci. Comput.* 1 (1983) 55–64.
- [75] P. Brown, G. Byrne, A. Hindmarsh, VODE: A variable-coefficient ODE solver, *SIAM J. Sci. Stat. Comput.* 10 (1989) 1038–1051.
- [76] N. Bou-Rabee, H. Owhadi, Long-run accuracy of variational integrators in the stochastic context, *SIAM J. Numer. Anal.* 48 (2010) 278–297.