To appear in *Molecular Physics*
Vol. 00, No. 00, Month 200x, 1–11

RESEARCH ARTICLE

Lattice-Boltzmann methods for suspensions of solid particles

Anthony J. C. Ladd*
Chemical Engineering Department, University of Florida,
Gainesville, FL 32611-6005, USA

(*Received 00 Month 200x; final version received 00 Month 200x*)

In this paper, dedicated to Prof. Jean-Pierre Hansen, I will summarize the development of lattice-Boltzmann methods for simulating the dynamics of colloidal suspensions. I will describe the main ideas and subsequent improvements, and place them in the wider context of particle-based methods for fluid dynamics.

1. Introduction

It is a pleasure and a privilege to contribute to this volume recognizing the important contributions of Prof. Jean-Pierre Hansen FRS to the field of molecular simulation. My first encounter with Jean-Pierre was through the marvelous (and at that time singular) textbook on molecular simulations he caoauthored with Ian McDonald [1]. At that time, part of my thesis work involved simulating three-phase coexistence (under the guidance of Les Woodcock); the seminal paper by Hansen and Verlet on the coexistence properties of the Lennard-Jones system [2] was a crucial reference. During that time I was able to meet and discuss the problem with Jean-Pierre at a NATO summer school near Ajaccio in Corsica, where he was one of the lecturers. Subsequently we developed a mutual interest in colloidal systems and have shared many enjoyable discussions over the years, beginning in Paris when he was at Jussieau and I would be attending CECAM workshops. Later we would meet in Lyon, where Jean-Pierre was instrumental in establishing and supporting a home for CECAM, and lastly in Cambridge. I have benefited much from his insight and knowledge over the years, and I wish him the best in this next phase of his career.

In this paper I will summarize the development of the lattice-Boltzmann method for simulating suspensions of solid particles, beginning with its origins in lattice-gas models for hydrodynamics [3]. A fundamental problem in modeling the dynamics of colloids is in determining the hydrodynamic interactions between the solid particles. These are the viscous (drag) forces due to the particle motion, and are hydrodynamic rather than thermodynamic in origin. The paper is in the nature of a personal perspective; for those interested in broader view of the problem there are at least three review articles available [4–6], as well as a more general review of lattice-Boltzmann methods [7] and the book by Sauro Succi [8], which will soon have a second edition available.

The standard method to simulate fluid flow is by a volume discretization of the partial differential equations describing the time evolution of the velocity and pres-

*Corresponding author. Email: tladd@che.ufl.edu

ISSN: 0040-5167 print/ISSN 1754-2278 online

DOI: 10.1080/0040516YYxxxxxxxx
http://www.informaworld.com

sure fields. If the mesh conforms to the particle surfaces this can give second-order (or higher) accuracy [9, 10], which is difficult to match with particle methods. However the challenges involved in reliably and efficiently moving the mesh have meant that alternatives methodologies have frequently been sought out, such as immersed boundary and related methods [11–13]. Nevertheless, object-oriented toolkits for dynamic mesh motion are now available in the public domain [14], which might make it more feasible to simulate suspensions of many particles with the (unstructured) mesh conforming to the particle shapes.

The most popular method of determining the hydrodynamic interactions between suspended particles is by a multipole representation of the force distribution on the particle surface [15]. There have been several numerical implementations of this idea [16–18], but the method is limited by the assumption of Stokes flow (zero Reynolds number) and the cubic scaling of the computational effort with the number of particles. Accelerated algorithms are possible, which are in principle of order N or $N \ln N$ (N is the number of particles), but the methods are complex and only achieve good scaling for large numbers of particles [19, 20].

It is important to recognize that a suspension is not a mixture, and attempts to model it with conservative dynamics are certain to come to grief. The issue is one of scales and makes its presence felt in two ways. First, in a suspension the rates of mass and momentum transfer are very different; the Schmidt number, the ratio of kinematic viscosity to diffusion, is of the order of 10^4 for polymer solutions (radius of gyration $\sim 10\,\mathrm{nm}$) and over 10^6 for colloidal particles. In a mixture these scales are roughly the same and the Schmidt number can only be increased by increasing the size ratio in proportion to the desired scale separation. The second problem with mixtures is one of depletion forces; the excluded volume interaction promotes an unphysical (from the point of view of a suspended particle) packing around the large sphere, whereas in reality the fluid should exert no thermodynamic force on the particle. A successful particle model for hydrodynamics must eliminate, or at least mitigate, these effects.

2. Particle models for continuum fluids

There is a long history of attempts to simulate fluid motion using particles, perhaps beginning with the realization that the decay of the velocity autocorrelation function in space and time tracked the predictions of incompressible fluid dynamics after a few collisions [21]. This is because any algorithm which conserves mass and momentum, is isotropic and Galilean invariant, will lead to hydrodynamic behavior at sufficiently long times. Perhaps the most successful particle model has been Smoothed Particle Hydrodynamics or SPH, developed for astrophysical applications in the 1970's by Gingold, Monaghan and Lucy. The key idea is to interpolate from a discrete set of variables (particles) to a differentiable field by using localized interpolating functions; Monaghan's review [22] is a good introduction to the method. Another pioneering idea came from simulating the flows of rarefied gases, where the Navier-Stokes equations break down. A direct solution of the Boltzmann transport equation (rather than a moment expansion of the collision kernel) is computationally demanding (a six-dimensional PDE) and Bird's solution was a stochastic simulation [23] that became known as the Direct Simulation Monte Carlo or DSMC method [24]. It shares the stream and collide structure of lattice-gas and stochastic rotation dynamics, but in this case the full Boltzmann collision operator is simulated, rather than just the hydrodynamically relevant moments. The system volume is divided into cells and pairs of particles within the same cell are randomly selected for collision until the correct number of events have taken place. Though the method is conceptually simple, it is easy to make mistakes in the sampling; fortunately there is extensive documentation of the pitfalls in Bird's books.

More recent particle-based methods for fluids simulations – lattice-gas (LG), lattice-Boltzmann (LB), Dissipative Particle Dynamics (DPD), and Stochastic Rotation Dynamics (SRD) – derive from a seminal paper suggesting how a lattice-gas could reproduce the equations of macroscopic hydrodynamics on sufficiently large scales [3]. Despite some encouraging early results [25], it soon became clear that lattice-gas models suffered from a number of serious drawbacks: the broken Galilean invariance could not be scaled away except in the case of single-phase flows, collision operators in three dimensions necessitated large lookup tables [26] or additional randomizing operations [27, 28], and the statistical fluctuations required a lot of averaging for a significant signal. The lattice-Boltzmann model was first introduced as a means to understanding the hydrodynamics of lattice gases [29], simplifying the problem by eliminating the statistical noise inherent in lattice-gas models. However, the lattice-Boltzmann model could only be turned into a viable computational algorithm by linearizing the Boltzmann collision operator [30], at the cost of the unconditional stability of models with the binary collisions [29]. The introduction of a linearized collision operator [30, 31] paved the way for a rather clever trick [32, 33], which introduced a low-velocity Galilean invariance into the lattice-Boltzmann model, making multiphase simulations possible.

A different approach to the lack of Galilean invariance is to lift the restriction that the particles must live on a lattice. Attempting to preserve some of the computational advantages of lattice gases, while restoring Galilean invariance gave rise to Dissipative Particle Dynamics or DPD [34]. DPD is similar to molecular dynamics, but with particles representing fluid packets rather than individual molecules. This leads to much softer forces than in typical MD simulations; in addition there is a dissipative friction between neighboring particles, balanced by random forces so as to satisfy the fluctuation-dissipation theorem. Theoretical analysis of DPD models [35] led to the eventual realization that the optimum DPD algorithm is the SPH method [22], with the addition of thermal fluctuations [36].

The other important particle simulation method to develop during this period was Stochastic Rotation Dynamics (SRD) [37] and its later incarnation Multi-Particle Collision Dynamics (MPC) [38]. The method is closely related to DSMC [23, 24], but instead of a stochastic simulation of the Boltzmann collision operator a much simpler and faster procedure is adopted, which is nevertheless sufficient for the hydrodynamic limit. Again the system is divided into cells, but all the particles within the cell undergo a collision simultaneously. The velocity vector of each particle relative to the center of mass velocity of the cell is rotated about a randomly chosen axis. This relaxes the non-equilibrium distribution while preserving mass and momentum conservation; Ref. [38] is an extensive review of the method.

Particle simulations of fluid motion remain an active area of investigation as the recent CECAM workshop (Particle-based modeling on meso-scales, Zaragoza, October 20-24 2014) shows. However, in the remainder of this article I will focus on lattice-Boltzmann methods for suspended particles.

3. The Boltzmann equation on a lattice - discretization of velocity space.

For a single species the state of the fluid is described by the mass density of particles n_i moving with a fixed velocity $\boldsymbol{c}_i$ labeled by the index i. The standard velocity sets for two and three dimensional flows [32] were originally developed heuristically, but were later found to correspond to the basis for a Gauss-Hermite quadrature over the velocity space [39, 40]. The lattice-Boltzmann algorithm can then be summarized by the equation

$$n_i(\boldsymbol{r} + \boldsymbol{c}_i h, t + h) = n_i(\boldsymbol{r}, t) + \Delta_i\left(\boldsymbol{n}(\boldsymbol{r}, t)\right). \tag{1}$$

The collision operator $\boldsymbol{\Delta}$ instantaneously converts the current velocity distribution $\boldsymbol{n}(\boldsymbol{r}, t)$ to the post-collision one $\boldsymbol{n}^\star = \boldsymbol{n} + \boldsymbol{\Delta}$; this distribution is then streamed to the neighboring nodes over the time step h. The velocity set $\boldsymbol{c}_i$ is chosen such that each new position $\boldsymbol{r} + \boldsymbol{c}_i h$ is again at a lattice site. The collision operator must conserve mass and momentum: $\sum_i \Delta_i = \sum_i \Delta_i \boldsymbol{c}_i = 0$.

Hydrodynamic fields – mass density ρ, momentum density $\boldsymbol{j}$, and momentum flux $\boldsymbol{\Pi}$ – can be defined as moments of the velocity distribution, as in the continuous Boltzmann equation:

$$\rho = \sum_i n_i, \tag{2}$$

$$\boldsymbol{j} = \sum_i n_i \boldsymbol{c}_i = \rho \boldsymbol{u}, \tag{3}$$

$$\boldsymbol{\Pi} = \sum_i n_i \boldsymbol{c}_i \boldsymbol{c}_i. \tag{4}$$

Conservation laws mean that any function of the mass and momentum densities $n^{eq}(\rho, \boldsymbol{j})$ is a collisional invariant, but there is flexibility in the functional form for n^{eq}. The original lattice-Boltzmann models [29–31] kept the equilibrium distribution derived from lattice-gases. However, although the statistical errors (and the effects of finite Schmidt number) were eliminated, Galilean invariance remained broken, manifesting itself in a density-dependent advection velocity and a velocity-dependent pressure. By taking a low-velocity expansion of the Maxwell-Boltzmann distribution (to quadratic order in the velocity),

$$n_i^{eq}(\rho, \boldsymbol{j}) = a^{c_i} \rho \left(1 + \frac{\boldsymbol{u} \cdot \boldsymbol{c}_i}{c_s^2} + \frac{(\boldsymbol{u} \cdot \boldsymbol{c}_i)^2}{2c_s^4} - \frac{u^2}{2c_s^2}\right), \tag{5}$$

the correct form for the Euler momentum flux is obtained [32, 33]:

$$\boldsymbol{\Pi}^{eq} = p\boldsymbol{1} + \rho \boldsymbol{u}\boldsymbol{u}, \tag{6}$$

where $p = \rho c_s^2$ is an ideal gas equation state with sound speed c_s. The weights a^{c_i} depend on the choice of LB model; for D3Q19, $a^0 = 1/3$, $a^1 = 1/18$, and $a^{\sqrt{2}} = 1/36$. Equation 5 is used in the majority of LB simulations, but other forms for the equilibrium distribution are possible, which can lead to improved stability [41, 42].

A key innovation that made lattice-Boltzmann a practical computational tool was the introduction of a linearized collision operator [30, 31], operating on the non-equilibrium distribution $\boldsymbol{n}^{neq} = \boldsymbol{n} - \boldsymbol{n}^{eq}$:

$$\Delta_i = \sum_j \mathcal{L}_{ij} n_j^{neq} + f_i, \tag{7}$$

where f_i may include an external force density or thermal fluctuations. The matrix $\mathcal{L}$ must conserve mass and momentum, and preserve isotropy up to fourth order, to correctly reproduce the Navier-Stokes equations; LB models for other phenomena (for example diffusion) may only require second order symmetry and fewer conservation laws. The simplest collision operator is the single-relaxation time or lattice-BGK model [32], which relaxes all the non-conserved degrees of freedom at the same rate: $\Delta_i = -n_i^{neq}/\tau$. However, it has been found that more flexible collision operators [33, 43] can lead to enhanced stability [44]. More significantly, the eigenvalues of the collision operator can tuned to improve the accuracy of the solution in the vicinity of solid boundaries [45]. It is usually more convenient to transform the densities n_i to moments in the velocity space, where the collision operator becomes diagonal and the symmetry relations more obvious [5].

External forces are introduced by constructing f_i so that there is an increase in momentum density $h\boldsymbol{f}$ over the course of a single time step, where $\boldsymbol{f}$ is the force density. The simplest approximation, $f_i = a^{c_i} h\boldsymbol{f} \cdot \boldsymbol{c}_i/c_s^2$, is frequently sufficient, although strictly speaking additional terms proportional to $\boldsymbol{uf}$ are also present [5, 46]. However, more important than the higher-order terms is the correct calculation of the momentum density, which, in the presence of a body force, is an average of the pre- and post-collision values [47, 48],

$$\boldsymbol{j} = \sum_i n_i \boldsymbol{c}_i + \frac{h}{2}\boldsymbol{f}. \tag{8}$$

The macroscopic behavior of lattice-Boltzmann models, including external forces and thermal fluctuations, can be obtained from a multi-scale expansion of Eq. (1) [5]. Lattice-Boltzmann models satisfy the equations of fluctuating hydrodynamics on sufficiently large length and time scales, with an ideal gas equation of state and a shear viscosity that is related to a specific eigenvalue of the collision operator.

The lattice-Boltzmann equation eliminates the thermal fluctuations intrinsic to lattice-gas models, but the noise can now be added back in a controlled way [49]. In LB models, stress is a dynamical variable with its own evolution equation, unlike conventional CFD where it is derived from gradients of the fluid velocity. This makes it straightforward to include thermal fluctuations by adding random noise to the stress tensor; momentum is automatically conserved and the amplitude of the noise can be related to the temperature [48] using the methods of fluctuating hydrodynamics [50]. However, thermalizing just the stress modes [48, 51], does not satisfy equipartition among the degrees of freedom [52], even though stress fluctuations are sufficient for the hydrodynamic limit. The simple but important advance is to thermalize all the (non-conserved) degrees of freedom, not just the stress. In this way the fluctuation-dissipation theorem is satisfied at all scales [52]. The insights from Ref. [52] have provoked further reflection, attempting to root the fluctuating lattice-Boltzmann equation (FLBE) more firmly within the framework of statistical mechanics [53, 54]. By introducing particles of variable mass at each lattice site (rather than mass densities), it becomes possible to control the thermal noise in a way that connects naturally to statistical mechanics; a small number of heavy particles have large fluctuations (for a fixed total mass) while a large number of light particles (with the same total mass) have small fluctuations [53]. Nevertheless, the broken Galilean invariance returns in the form of velocity dependent correlation matrix, which was ignored in Ref. [53]. Recent work has shown how the velocity-independent correlations can be introduced into FLBE simulations [54], minimizing the lattice artifacts for a small computational penalty, although the algorithm is no longer simple. This remains an area of ongoing research.

Prof. Hansen has, perhaps wisely, remained largely in the more well founded arena of molecular simulation, but he has been involved in the development of lattice kinetic models for the Fokker-Planck equation [55]. A discrete kinetic equation was found by applying a polynomial basis to the continuum Fokker-Plank equation, just as had been done for the lattice-Boltzmann equation [39, 40]. The connection between the microscopic and macroscopic scales was then accomplished with a Chapman-Enskog expansion. It would be interesting to explore the connection between the fluctuating lattice-Boltzmann model and the lattice Fokker-Planck equation; it would seem that the former is a stochastic realization of the latter.

4. Including solid particles

In order to simulate particle suspensions the motion of particles and fluid must be coupled together. The standard hydrodynamic boundary condition, valid on scales larger then a few tens of nanometers (except in rarefied gases) is the no-slip condition, where each element of the fluid near a boundary moves with the local velocity of the adjacent particle surface. Boundary conditions in the lattice-Boltzmann method can be implemented by modifications to the distributions encountering the surface. Although the basic implementation of a no-slip boundary condition is rather simple [48], a large literature has developed devoted to improvements in accuracy and stability; a summary of work up to about 2009 can be found in two review articles [5, 6]. Here I will attempt to give a framework and context for this work and comment on some technical issues that sometime pass unnoticed.

Let us begin with the simplest case; a planar solid boundary aligned with the grid and placed half-way between lattice planes. A no-slip boundary can be implemented by the "bounce-back" (BB) rule [48]

$$n_{i'}(\boldsymbol{r}, t+h) = n_i(\boldsymbol{r}, t) - \frac{2a^{c_i}\rho \boldsymbol{u}_b \cdot \boldsymbol{c}_i}{c_s^2}, \tag{9}$$

where $\boldsymbol{u}_b$ is the local velocity of the boundary surface. Particle forces and torques can be calculated from the momentum transfered to the fluid at each boundary node. The value of this simple rule, which emerged from earlier work on lattice-gas models [27, 28], is that it can be applied to any surface, just by finding all the intersecting links. It leads to a first-order accurate boundary condition, which means that the hydrodynamic boundary (where the velocity vanishes) is offset from the physical boundary by a distance of order of the spacing between lattice nodes. Despite its limitations, it remains one of the most commonly used boundary conditions in simulations of suspensions, because of its simplicity and robustness.

There has been a fairly widespread misinterpretation of the physics of the solid-fluid boundary condition when there is motion normal to the solid surface. In that case Eq. (9) predicts that mass crosses the solid boundary in what appears to be an unphysical manner, and corrections have been derived to maintain exact mass conservation in the fluid region [56, 57]. However, this correction fails to take account of the motion of the particle itself and is therefore incorrect. In essence the dynamics of a suspension (as usually implemented) is a form of operator splitting, alternating between particle and fluid updates [5]; a proper accounting of the mass balance can only take place when both phases have moved. For example, with a planar wall, there is one link incident in the direction normal to the surface ($a^1 = 1/18$) and 4 links incident at 45° to the normal ($a^{\sqrt{2}} = 1/36$). Adding these contributions, we find the mass flux per lattice node is $-\rho u_n$, where u_n is the component of $\boldsymbol{u}_b$ normal to the surface. Thus a slab of fluid of thickness $u_b h$ crosses the boundary during the fluid motion part of the update, but the particle moves

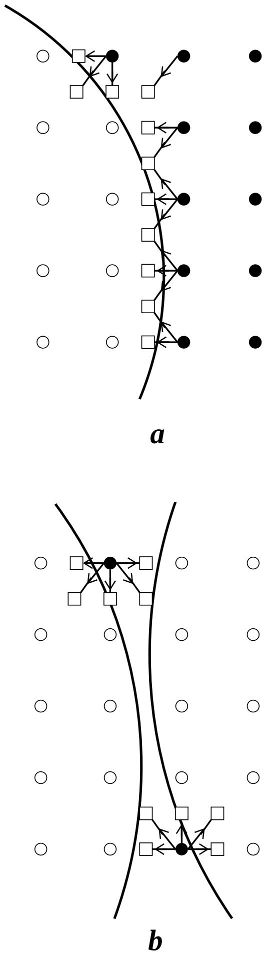

Figure 1. Location of boundary nodes for a curved surface (a) and two surfaces in near contact (b). The velocities along links cutting the boundary surface are indicated by arrows. The locations of the boundary nodes are shown by open squares, and the fluid nodes by solid circles.

the same distance in the next part of the update, restoring the fluid to its original position with respect to the solid surface. Adding additional terms to prevent fluid crossing the particle surface during the fluid update leads to an artificial pressure drop across a solid particle, with a net compression behind and a rarefaction in front.

A possible drawback of LB models is that the location of the hydrodynamic boundary is affected by the eigenvalues of the collision operator. However, this variability can instead be utilized to make the hydrodynamic boundary coincide with the physical one [45]. Unfortunately the ideal tuning varies with both location and orientation (with respect to the grid), even for a planar boundary. Thus in the general case, where many different positions and orientations are sampled, the tuning must be an effective one, designed to produce the best overall result but not necessarily exact at each point. Originally the hydrodynamic radius of a particle had to be determined empirically for every viscosity [48], but the ability to tune the behavior of the collision operator near the boundary has made that unnecessary, even with the bounce-back condition.

In order to have a more accurate (second-order) boundary condition, particularly when there are curved surfaces (Fig. 1), some form of interpolation must be used [58–60], which breaks the exact mass conservation of the bounce-back scheme but leads to a significant improvement in the accuracy and smoothness of the flow fields around a solid particle. There have been a large number of interpolation schemes proposed, most of them related to the multireflection (MR) boundary condition [45] or some simplification derived from it. These methods include additional lattice points along the $\boldsymbol{c}_i$ direction as a basis for interpolation: $n_i(\boldsymbol{r}, t)$ (as in the BB scheme), $n_i(\boldsymbol{r} - h\boldsymbol{c}_i, t)$ in the simplest linear interpolation schemes and $n_{i'}(\boldsymbol{r}, t)$, $n_{i'}(\boldsymbol{r} - h\boldsymbol{c}_i, t)$ and $n_i(\boldsymbol{r} - 2h\boldsymbol{c}_i, t)$ in the most general MR scheme. There is a range of choices in the coefficients, which can be systematically adjusted to eliminate higher-order errors in the boundary conditions [61, 62].

Besides the loss of mass conservation, the increased stencil (3 points for MR versus 1 for BB) is a drawback when there are narrow gaps between the solid surfaces. The problem of the larger stencil can be somewhat alleviated by using the equilibrium distribution on the surface as another interpolation point [63], but mass conservation is not restored. Mass conservation can be restored by including a wider range of velocities [64], but such methods (see also Ref. [65]) have not been popular because of the difficulties in applying them to complex geometries, where there are variable numbers of incoming velocities.

The advancements in implementing boundary conditions outlined in the preceding paragraphs enable second-order accurate calculations of the flow field in a wide variety of geometries to be routinely and straightforwardly made. However, for suspended particles it is also necessary to calculate the forces and torques on the particles. For rigid particles the standard momentum exchange method [48] is adequate and can readily be adapted to higher-order boundary conditions [66]. However the forces vary significantly across the boundary surface [67], in large part because the velocity vectors point in different directions. It therefore becomes necessary to average the force over a small region in order to obtain a spatially smoother stress on the boundary, which introduces another layer of complexity and an additional parameter, the size of the patch over which the forces are averaged. Analysis suggests that the optimum patch size scales as the square root of the grid spacing, so that the number of points in the patch increases in proportion to the linear resolution [67]; in this case the error in the local force also scales roughly as the square root of the grid spacing. An alternative to the momentum exchange is to integrate the fluid stress over the surface of the particle [68], but this is both cumbersome and less accurate on the global scale than momentum exchange [69], although the local stresses are much smoother. In summary, the implementation of boundary conditions in lattice-Boltzmann models remains elegant and has become increasingly accurate and in some instances simpler to implement [70]. However, evaluation of the force distribution remains problematic; this has limited the application of these methods to rigid particles.

It is frequently the case that the interior of a solid particle remains filled with fluid [48, 51], which is a slightly different physics from a rigid particle. On the viscous time scale (typically 10^{-6} s) the interior fluid contributes an additional inertia to the particle motion, which has a negligible effect on the dynamics [51]. However at very short times, when inertial effects are significant, there can be noticeable oscillations in the particle velocity [71]. The effects of the interior fluid can be eliminated in a number of different ways [57, 71, 72]; however, the one-side boundary condition (without interior fluid), illustrated in Fig 1 requires an additional correction in the presence of an external shear flow [73]. Moreover, changes in local fluid momentum due to sites being covered or uncovered must be included in the surface forces to ensure overall momentum conservation. Including interior fluid has the advantage that fluid does not need to be created or destroyed as nodes are covered or uncovered by the translating surface. Significantly smoother motion is obtained if the non-equilibrium distribution at the uncovered sites is included (by extrapolation from neighboring fluid sites) [66, 73].

There are a number of technical problems that arise when rigid particles come into near contact [72, 74], and in particular the treatment of lubrication forces. When two particles get close to one another the nature of the interaction changes and the forces are dominated by fluid flow in the narrow gap between the particles. Since this flow cannot be resolved on a fixed grid an alternative approximation to the force is needed. One possibility is to patch in the lubrication forces at short distances [72]. This ensures that the short-range interactions are well accounted for, although there is typically a region near the patch point where the force is less accurate. However, it is a bit clumsy to implement and typically requires an implicit solution for stability; in dense suspensions this can be problematic with many particles with common surfaces in near contact linked in one large cluster. Alternatively an intuitive but not entirely accurate procedure has been developed to calculate lubrication forces along the links [74]. This is a flexible scheme that provides an additional resistance at small distances, but it is not a quantitative representation of the lubrication forces.

Immersed boundary methods (IBM), which were first introduced more than 40 years ago [75] for modeling blood flow in the heart, are an important alternative for soft matter simulations The key idea is to solve the fluid and particle equations on different grids; the fluid on a fixed (Eulerian) grid and motion of the solid-fluid boundaries on a moving (Lagrangian) grid. The Lagrangian points move freely through the fluid, exchanging momentum with neighboring fluid sites; they can represent both rigid [11, 12, 76, 77] and deformable particles [78, 79]. The fluid equations, which typically make up the bulk of the computational effort, can be solved on a structured grid, either by conventional CFD methods [12, 13, 76, 77, 80] or by lattice Boltzmann [6, 81]. A drawback of immersed boundary methods is that interface is not sharp but spread out over a few grid cells. However, a regular grid makes it particularly attractive to use adaptive meshes [82], minimizing the error introduced by the slightly diffuse interface. Adaptive meshes can be used with lattice-Boltzmann methods as well [83]. Lubrication forces would still be necessary to simulate the hydrodynamic interactions of particles near contact.

5. Conclusion

The lattice-Boltzmann method has experienced exponential growth over the last 20 years and it looks as if it will continue to be an important computational method for fluid flows. With regard to particle suspensions, the immersed boundary method seems more promising, particularly for deformable particles, than methods based on imposition of solid-fluid boundary conditions. The distribution of forces in space and time is much smoother than typical implementations using bounce-back or higher-order boundary conditions. In my view conventional finite-volume methods are also potentially interesting. They have the advantage of a near exact representation of the particle surface, with exact conservation of fluxes between cells. The drawback of such methods, namely the complexity of creating and updating an unstructured mesh, has been mitigated by the introduction of advanced meshing tools such as the OpenFOAM® toolkit. It remains to be seen if such methods can be competitive with LB/IMB based simulations.

References

[1] J.P. Hansen and I.R. McDonald, *Theory of Simple Liquids* (Academic Press, London, 1986).
[2] J.P. Hansen and L. Verlet, Phys. Rev. **184**, 151 (1969).

[3] U. Frisch, B. Hasslacher and Y. Pomeau, Phys. Rev. Lett. **56**, 1505 (1986).

[4] A.J.C. Ladd and R. Verberg, J. Stat. Phys. **104**, 1191 (2001).

[5] B. Dünweg and A.J.C. Ladd, Adv. Polym. Sci. **221**, 89 (2009).

[6] C.K. Aidun and J.R. Clausen, Annual Review of Fluid Mechanics **42**, 439 (2010).

[7] S. Chen and G.D. Doolen, Ann. Rev. Fluid Mech. (30), 329 (1998).

[8] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond* (Oxford University Press, New York, 2001).

[9] J. Feng, H.H. Hu and D.D. Joseph, J. Fluid Mech. **261**, 95 (1994).

[10] J. Feng, H.H. Hu and D.D. Joseph, J. Fluid Mech. **277**, 271 (1994).

[11] A.L. Fogelson and C.S. Peskin, J. Comput. Phys. **79**, 50 (1988).

[12] N.A. Patankar, P. Singh, D.D. Joseph, R. Glowinski and T.W. Pan, Int. J. Multiphase Flow **26**, 1509 (2000).

[13] C.S. Peskin, Acta Numerica **11**, 479 (2002).

[14] H. Weller, G. Tabor, H. Jasak and C. Fureby, Comput. Phys. **12**, 620 (1998).

[15] P. Mazur and W.V. Saarloos, Physica A **115**, 21 (1982).

[16] J.F. Brady and G. Bossis, Ann. Rev. Fluid. Mech. **20**, 111 (1988).

[17] A.J.C. Ladd, J. Chem. Phys. **88**, 5051 (1988).

[18] B. Cichocki, B.U. Felderhof, K. Hinsen, E. Wajnryb and J. Blawdziewicz, J. Chem. Phys. **100**, 3780 (1994).

[19] A.S. Sangani and G.B. Mo, Phys. Fluids **8**, 1990 (1996).

[20] A. Sierou and J.F. Brady, J. Fluid Mech. **448**, 115 (2001).

[21] B.J. Alder and T.E. Wainwright, Phys. Rev. A **1**, 18 (1970).

[22] J.J. Monaghan, Annu. Rev. Astron. Astr. **30**, 543 (1992).

[23] G.A. Bird, *Molecular Gas Dynamics* (Oxford University Press, London, 1976).

[24] G.A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows* (Oxford University Press, Oxford, 1994).

[25] D. d'Humières, CR Acad Sci IIB **301**, 1391 (1985).

[26] B. Dubrulle, U. Frisch, M. Hénon and J.P. Rivet, Physica D **47**, 27 (1991).

[27] A.J.C. Ladd and D. Frenkel, Dynamics of colloidal dispersions via lattice-gas models of an incompressible fluid. in *Cellular Automata and Modeling of Complex Physical Systems*, edited by P. Manneville, N. Boccara, G. Y. Vichniac and R. Bidaux, no. 46 in Springer Proceedings in Physics (Springer-Verlag, Berlin-Heidelberg, 1989), pp. 242–245.

[28] A.J.C. Ladd and D. Frenkel, Physics of Fluids A **2**, 1921 (1990).

[29] G.R. McNamara and G. Zanetti, Phys. Rev. Lett. **61**, 2332 (1988).

[30] F. Higuera, S. Succi and R. Benzi, Europhys. Lett. **9**, 345 (1989).

[31] S. Succi, E. Foti and F. Higuera, Europhysics Lett. **10** (5), 433 (1989).

[32] Y.H. Qian, D. d'Humières and P. Lallemand, Europhys. Lett. **17**, 479 (1992).

[33] G.R. McNamara and B.J. Alder, Physica A **194**, 218 (1993).

[34] P.J. Hoogerbrugge and J.M.V.A. Koelman, Europhys. Lett. **19**, 155 (1992).

[35] P. Espanol and P. Warren, Europhys. Lett. **30**, 191 (1995).

[36] P. Espanol and M. Revenga, Phys. Rev. E **67** (2003).

[37] A. Malevanets and R. Kapral, J. Chem. Phys. **110**, 8605 (1999).

[38] G. Gompper, T. Ihle, D.M. Kroll and R.G. Winkler, Advan. Polym. Sci. **221**, 1 (2009).

[39] X. He and L.S. Luo, Phys. Rev. E **55**, R6333 (1997).

[40] X. He and L.S. Luo, Phys. Rev. E **56**, 6811 (1997).

[41] I.V. Karlin, A. Ferrante and H.C. Öttinger, Europhys. Lett. **47**, 182 (1999).

[42] B.M. Boghosian, J. Yepez, P.V. Coveney and A. Wager, Proc. Roy. Soc. A-Math Phy **457** (2007), 717 (2001).

[43] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand and L.S. Luo, Phil. Trans. Royal Soc. London A **360**, 437 (2002).

[44] P. Lallemand and L.S. Luo, Phys. Rev. E **61**, 6546 (2000).

[45] I. Ginzburg and D. d'Humieres, Phys. Rev. E **68** (6, Part 2), 066614 (2003).

[46] Z. Guo, C. Zheng and B. B. Shi, Phys. Rev. E **65**, 046308 (2002).

[47] I. Ginzburg and P.M. Adler, J. Phys. II France **4**, 191 (1994).

[48] A.J.C. Ladd, J. Fluid Mech. **271**, 285 (1994).

[49] A.J.C. Ladd, Phys. Rev. Lett. **70**, 1339 (1993).

[50] L.D. Landau and E.M. Lifshitz, *Fluid Mechanics* (Addison-Wesley, London, 1959).

[51] A.J.C. Ladd, J. Fluid Mech. **271**, 311 (1994).

[52] R. Adhikari, K. Stratford, M.E. Cates and A.J. Wagner, Europhys. Lett. **71**, 473 (2005).

[53] B. Dünweg, U.D. Schiller and A.J.C. Ladd, Phys. Rev. E **76**, 036704 (2007).

[54] G. Kaehler and A.J. Wagner, Phys. Rev. E **87** (6), 063310 (2013).

[55] D. Moroni, B. Rotenberg, J.P. Hansen, S. Succi and S. Melchionna, Phys. Rev. E **73** (6),

066707 (2006).

[56] C.K. Aidun and Y.N. Lu, J. Stat. Phys. **81**, 49 (1995).

[57] C.K. Aidun, Y.N. Lu and E. Ding, J. Fluid Mech. **373**, 287 (1998).

[58] O. Filippova and D. Hänel, J. Comput. Phys. **147**, 219 (1998).

[59] R.W. Mei, L.S. Luo and W. Shyy, J. Comput. Phys. **155**, 307 (1999).

[60] M. Bouzidi, M. Firdaouss and P. Lallemand, Phys. Fluids **13**, 3452 (2001).

[61] I. Ginzburg, Adv. Water Res. **28** (11), 1171 (2005).

[62] I. Ginzburg, Adv. Water Res. **28** (11), 1196 (2005).

[63] B. Chun and A.J.C. Ladd, Phys. Rev. E **75**, 066705 (2007).

[64] J. Bao, P. Yuan and L. Schaefer, J. Comput. Phys. **227**, 8472 (2008).

[65] D.R. Noble, S.Y. Chen, J.G. Georgiadis and R.O. Buckius, Phys. Fluids **7**, 203 (1995).

[66] P. Lallemand and L.S. Luo, J. Comp. Phys. **184** (2), 406 (2003).

[67] A. Caiazzo and M. Junk, Comput. Math. Appl. **55** (7), 1415 (2008), Mesoscopic Methods in Engineering and Science.

[68] X. He and G. Doolen, J. Comput. Phys. **134**, 306 (1997).

[69] R. Mei, D. Yu, W. Shyy and L.S. Luo, Phys. Rev. E **65**, 041203 (2002).

[70] X. Yin and J. Zhang, J. Comput. Phys. **231**, 4295 (2012).

[71] M.W. Heemels, M.H.J. Hagen and C.P. Lowe, J. Comput. Phys. **164**, 48 (2000).

[72] N.Q. Nguyen and A.J.C. Ladd, Phys. Rev. E **66**, 046708 (2002).

[73] E. Lorenz, A. Caiazzo and A.G. Hoekstra, Phys. Rev. E **79** (3), 036705 (2009).

[74] E.J. Ding and C.K. Aidun, J. Stat. Phys. **112**, 685 (2003).

[75] C.S. Peskin, J. Computational Phys. **10** (2), 252 (1972).

[76] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph and J. Periaux, Comput. Method Appl Math Engng **184**, 241 (2000).

[77] T.W. Pan and R. Glowinski, J. Comput. Phys. **181**, 260 (2002).

[78] J.F. Zhang, P.C. Johnson and A.S. Popel, J. Biomechanics **41** (1), 47 (2008).

[79] J.S. Wu and C.K. Aidun, Int. J. For Numerical Methods In Fluids **62** (7), 765 (2010).

[80] C.S. Peskin and D.M. McQueen, J. Computational Phys. **81** (2), 372 (1989).

[81] Z.G. Feng and E.E. Michaelides, Journal of Computational Physics **195** (2), 602 (2004).

[82] A.M. Roma, C.S. Peskin and M.J. Berger, J. Comput. Phys. **153** (2), 509 (1999).

[83] D.Z. Yu, R.W. Mei and W. Shyy, Int. J. Numer. Math. Fluids **39** (2), 99 (2002).