Simulating colloid hydrodynamics with lattice Boltzmann methods

Simulating colloid hydrodynamics with lattice Boltzmann methods

M E Cates[1]**, K Stratford**[2]**, R Adhikari**[1]**, P Stansell**[1]**, J-C Desplat**[2]**,
I Pagonabarraga**[3] **and A J Wagner**[4]

[1] School of Physics, JCMB Kings Buildings, The University of Edinburgh, Mayfield Road,
Edinburgh EH9 3JZ, UK
[2] EPCC, JCMB Kings Buildings, The University of Edinburgh, Mayfield Road,
Edinburgh EH9 3JZ, UK
[3] Departament de Fisica Fonamental, Universitat de Barcelona, Avenida Diagonal 647,
08028 Barcelona, Spain
[4] Department of Physics, North Dakota State University, ND 58105, USA

Abstract
We present a progress report on our work on lattice Boltzmann methods for
colloidal suspensions. We focus on the treatment of colloidal particles in binary
solvents and on the inclusion of thermal noise. For a benchmark problem of
colloids sedimenting and becoming trapped by capillary forces at a horizontal
interface between two fluids, we discuss the criteria for parameter selection,
and address the inevitable compromise between computational resources and
simulation accuracy.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

The lattice Boltzmann equation (LB equation, or LBE) is a widely used lattice formulation
of fluid mechanics [1]. It offers a faithful discretization of the Navier–Stokes equation of
isothermal, near-incompressible fluid flow, and is very well adapted to parallel computation [2].
Although used for large-scale fluid dynamics simulations such as those of flows around aircraft
wings [3], the LBE approach is particularly adapted to simulating mesoscopic problems [4].
These include, for example, porous medium flows, and flows of complex and multicomponent
fluids with microstructure [5–9]. The latter can be modelled using various extensions of the
basic algorithm for a single-component fluid [9–11].

In this paper we outline some recent progress towards the creation and use of a versatile
LB code for colloids in single-phase and (particularly) biphasic fluid solvents [12]. The LB
algorithm is not wholly intuitive, and the mapping of simulation parameters onto real ones has
to be carried out with some care, with attention paid to possible sources of systematic error.

For problems involving coarsening of binary fluids, the required validation is provided in [6]; below, we discuss some of the additional considerations that arise for colloids (section 6). In addition, because of the mesoscopic length scales involved, such a code must allow a proper treatment of thermal noise. We have recently identified and resolved a long-standing difficulty with the incorporation of noise in the LB approach [13], and this is discussed in section 7. Before turning to these topics we briefly review, in sections 2–5, the LB algorithm for a single-phase fluid, for binary fluids, and for colloids.

2. The LB algorithm for a single-phase fluid

The LB approach works one level beneath the usual equations of hydrodynamics, at the level of the collisional and propagating dynamics of distribution functions. The distribution function $f_i(\mathbf{x}, t)$ at lattice site $\mathbf{x}$ can be thought of as the density of (fictitious) fluid particles with velocity $\mathbf{c}_i$ at this site. The velocity set is chosen so that in one time step Δt, the displacement $\mathbf{c}_i \Delta t$ represents either the displacement to a neighbouring lattice site or the null displacement. We use a cubic D3Q15 lattice, meaning a three-dimensional lattice with 15 velocities at each site [15].

Note that a larger number of velocities, and hence of distribution functions, are introduced at each lattice site than is strictly needed to create the correct number of hydrodynamic degrees of freedom for fluid-mechanical purposes. These extra degrees of freedom are required so as to ensure rotational and Galilean invariance [1]; they lead to 'ghost modes' which feature strongly in section 7 below. Setting these ghost modes aside, the hydrodynamic fields (which are all that matter for athermal fluid mechanics) are represented by various moments of the distribution functions. For example the local density of the fluid is the zeroth moment

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t) \tag{1}$$

whereas the local momentum density $\mathbf{g} = \rho \mathbf{v}$ is given by

$$\rho \mathbf{v} = \sum_i f_i(\mathbf{x}, t) \mathbf{c}_i \tag{2}$$

and the momentum flux (or kinetic stress tensor) is given by

$$\mathbf{\Pi} = \sum_i f_i \mathbf{c}_i \mathbf{c}_i \tag{3}$$

where $\mathbf{c}_i \mathbf{c}_i$ is a dyadic product.

Although the LB method actually describes a compressible fluid, in using it one always ensures that the Mach number is kept small with the result that the flow is nearly incompressible. We choose our unit of mass so that $\rho = 1$ for a quiescent fluid, and choose the lattice parameter as the unit of length, and the time step as the unit of time ($\Delta t = 1$). This defines a set of lattice units (LU). The interconversion between these and real physical units raises interesting issues in parameter selection (see section 6).

The LB algorithm updates the distribution functions f_i by a combination of 'streaming' and 'collision'. These are often thought of as two different steps in the algorithm. Streaming passes each distribution function f_i to the neighbouring site appropriate to the velocity $\mathbf{c}_i$ that it governs; whereas collision is an on-site update of the various f_i which conserves mass and momentum (but not energy), creates dissipation, and allows diffusion of momentum. (The diffusivity of momentum is the kinematic viscosity η/ρ of the fluid.) In combination, the streaming and collision steps may be written as

$$f_i(\mathbf{x} + \mathbf{c}_i, t + 1) - f_i(\mathbf{x}, t) = \sum_j L_{ij} \left(f_j(\mathbf{x}, t) - f_j^0(\mathbf{x}, t) \right) \tag{4}$$

where L_{ij} is a collision matrix, often chosen as the lattice BGK matrix δ_{ij}/τ with τ a relaxation time. In equation (4), $f_j^0(\mathbf{x}, t)$ is an equilibrium distribution [1] which itself depends on the local values of $\rho(\mathbf{x}, t)$ and $\mathbf{v}(\mathbf{x}, t)$, so the collision process (even with the stated BGK form for L_{ij}) is not diagonal among the f s. (The nontrivial structure of its eigenmodes will be important in section 7.)

Meanwhile, the fluid shear viscosity is given (in LU) by

$$\eta = c_s^2(2\tau - 1)\rho/2 \tag{5}$$

where c_s is the sound speed; for the D3Q15 lattice that we use, $c_s = 1/\sqrt{3}$ in LU. (There is also a bulk viscosity, which for this lattice is $\zeta = 2\eta/3$. Since the code is run in a near-incompressible limit, the precise value of the bulk viscosity is relatively unimportant.) In practice $\tau = 1$ is numerically efficient: in this case the collision resets the f_i to local equilibrium, each time step. Larger values of τ are possible, but can lead to numerical problems. In particular it is not possible for the momentum to be transported across a given distance by diffusion any faster than it can get there via sound modes (at speed c_s), so η/ρ values large compared to c_s will give incorrect momentum transport at short distances (and/or unwanted non-Newtonian effects [16]). Viscosity values much smaller than unity can be used, and these are very helpful in studying phase separation of binary fluids at high Reynolds number [6, 7].

The basic LB algorithm, as just described, is sufficient to model isothermal flow of a single-phase fluid with a variety of boundary conditions [1]. In the continuum limit for the bulk fluid, one recovers (with Greek suffices for Cartesian components)

$$\partial_t \rho + \nabla_\alpha g_\alpha = 0 \tag{6}$$

$$\partial_t g_\alpha + \nabla_\beta \Pi_{\alpha\beta} = 0 \tag{7}$$

$$\Pi_{\alpha\beta} = g_\alpha v_\beta + p\delta_{\alpha\beta} - \eta_{\alpha\beta\gamma\epsilon}\nabla_\gamma v_\epsilon. \tag{8}$$

Here $v_\alpha = g_\alpha/\rho$ is the local fluid velocity, p is the pressure in a quiescent fluid (given in the LB approach by an ideal gas equation of state $p = \rho c_s^2$), and $\eta_{\alpha\beta\gamma\epsilon} = \eta\delta_{\alpha\gamma}\delta_{\beta\epsilon} + \eta\delta_{\beta\gamma}\delta_{\alpha\epsilon} + \zeta\delta_{\alpha\beta}\delta_{\gamma\epsilon}$ is the tensor of viscosities appropriate to a Newtonian fluid. It is possible to impose shearing boundary conditions through an imposed fluid velocity or stress at a pair of walls parallel to lattice directions; less obviously (given the existence of the underlying lattice) one can also introduce Lees–Edwards-type sheared periodic boundary conditions [14]. One limitation is that the fluid velocity throughout the system must remain small compared to the sound speed. This is a stronger condition than that of small Mach number (small compressibility) because it implies an absolute reference frame against which the flow speed is limited. In shear flow, however, this limit can be overcome by judicious use of multiple Lees–Edwards planes [14].

3. Binary fluids in the LB approach

Binary fluids can be handled by an extension to the above approach [10, 15], in which a second set of distribution functions $g_i(\mathbf{x}, t)$ is introduced. Analogously with equations (1)–(3), the low-order moments govern the compositional order parameter $\phi(\mathbf{x}, t)$ and its advective and diffusive fluxes. The equilibrium distribution g_i^0 involves an order parameter mobility and a chemical potential μ which derives from a well-chosen free energy functional $F(\phi, \nabla\phi)$, which we take to be of Landau–Ginzburg form [6, 15]. A second relaxation time is also introduced although, rather unintuitively, the order parameter mobility, as long as it is independent of ϕ, can be varied without changing this relaxation time (varying it instead by the choice of g_i^0). In our codes this second relaxation time is set to unity to optimize numerical efficiency; thus the g_i are reset to local equilibrium every time step.

The binary fluid LB algorithm is in many ways less satisfactory than that for a single fluid. For example, if a static droplet of one fluid is surrounded by another, there are weak violations of detailed balance due to 'spurious fluxes' which arise from the fact that the proper conservation laws for the order parameter are not built in at a deep enough level. In future it may be possible to improve on this [17], but meanwhile the algorithm gives useful results for phase separation dynamics [6, 7, 18, 19], phase separation with a temperature ramp [20], droplet break-up under shear [21, 22], and related problems. In combination with Lees–Edwards boundary conditions it allows the problem of sheared spinodal phase separation to be studied; we hope to publish more on this soon [23].

One advantage of the LB method over others for binary fluids is that, by working at the level of distribution functions, it avoids the hydrodynamic singularities that arise during pinch-off and topological reorganization of fluid domains [24]. Such singularities are smoothed out by order parameter diffusion; obviously this is only helpful under conditions where the singularities are a nuisance, rather than the controlling physical effect. The binary fluid LB method does require careful parameter steering to avoid spurious effects such as anisotropy of the fluid interface, and excessive order parameter diffusion in regions where this should be negligible compared to advection [6].

4. Colloid hydrodynamics in the LB approach

The introduction of moving solid objects in the LB method is a somewhat complicated procedure [9, 25–27]. Firstly the objects have to be mapped onto the lattice grid; but since they may be large and slowly moving, their positions cannot be moved by discrete jumps of one lattice unit. Hence the colloids are modelled off lattice, but each defines a set of links of the lattice crossed by the surface of the colloidal particle, and this link set is subject to discrete modification at each time step. The fluid distribution functions at these link nodes is handled via a 'bounce-back' procedure, in which velocities that would correspond to crossing into the colloid are reflected back into the fluid according to certain rules. The resulting force and torque on the colloid is found, and used to update its velocity and angular velocity. This information is not only used to update the colloid positions each time step, but fed back into the bounce-back prescription for the next step (which depends on the local velocity of the moving solid boundary).

In our work we adopt and modify the prescription of [27]. In contrast to earlier work [9] where the interior of each colloid was filled with a fictitious fluid, here the bounce-back rules treat the colloidal particles as truly solid. This distinction is important when dealing with binary solvents, in which one needs to develop rules for the compositional order parameter ϕ as well as the other hydrodynamic fields [12]. Due to the changing discretization of the boundary links, the shape of each colloid is in effect changing slightly as it moves across the lattice; but it is possible to calibrate this effect and get acceptable hydrodynamic behaviour for surprisingly small colloids, for example of radius $a = 2.4$ lattice units. However, the discrepancy between the real particle radius and the measured hydrodynamic one increases with viscosity [12], for reasons that are currently under investigation.

An alternative route would be to treat colloids as point particles and couple each of these locally to the fluid velocity, with a friction constant fixed by the Stokes law. This is a very useful approach to polymer hydrodynamics, where the procedure is applied to each bead in a long chain [28]. Since that problem is known to be dominated by far-field effects (leading to Zimm rather than Rouse dynamics), the fact that the near-field flow is treated inaccurately by such an approach does not matter. However, for colloid hydrodynamics it certainly does matter—for example, this method would not give accurate results for sedimentation of a small

group of particles at separations comparable to their diameters, even if no very close contacts (lubrication forces) came into play. Hence for a general-purpose colloidal hydrodynamics code there is no way to bypass the bounce-back procedure, or some equivalently elaborate scheme, in which forces are explicitly distributed over the surface of each colloid.

An important part of the hydrodynamics of interacting colloids involves near-field lubrication forces. At small surface-to-surface distances these have a strong divergence in the normal component (and a much weaker one in the tangential one) which the LB method can only resolve down to separations between colloidal surfaces of order one lattice spacing. These strong local contributions, unlike the other hydrodynamic terms, are pairwise additive across the colloid–colloid contacts. Hence their absence from the LB method can be rectified by patching in, at short distances $r < r_c$ only, a velocity-dependent lubrication force [27], acting directly between the particles. (In our code, we so far treat the normal component only [12]; r_c is chosen empirically, and corresponds to a lubrication layer thickness $h = r - 2a$ of about 0.5 in lattice units.) However, this brings in its own difficulties; because the force is velocity dependent, an implicit update scheme is now essential for the colloid dynamics. The computational time required for this scales badly with the size of any 'hydrodynamic clusters' (clusters of colloids mutually linked together by separations less than r_c) and such clusters can get very large at high concentrations.

In effect, in this regime, one is inverting the pairwise-additive colloidal drag matrix to get a mobility matrix, every time step. Indeed algorithms for colloid hydrodynamics exist that do only this, within a pairwise lubrication approximation [29], ignoring the far-field contributions that in our case are handled by the LB method. (This far field can be handled within Stokesian dynamics [30] which, like the pairwise lubrication algorithm but unlike the LB one, assumes creeping flow.) Such work shows that deviations from the hard-sphere potential are critical in determining, for example, jamming behaviour under shear [29].

In principle an LB algorithm with lubrication corrections should give results that can encompass this strong clustering limit without sacrificing the long-distance aspects of many-body hydrodynamics [27]. However, in view of the computational scaling issue, we have not attempted to explore this aspect, and prefer instead to bypass the lubrication problem entirely. This can be done by introducing a strong, short-range repulsion between colloids designed to ensure that interparticle separations $r < r_c$ are rare, so that the bad scaling does not arise. Of course such a force is often actually present, for example, in colloids interacting via a screened Coulomb force. We expect this additional repulsion to be unimportant for some scientific issues (including those that we address below) but not others (e.g. not hydrodynamic jamming under shear [29]).

5. Colloids in binary solvents

A colloid in a pair of solvents is said to exhibit neutral wetting when the solid–fluid interfacial tensions $\tilde{\sigma}$ are the same for the two solvents. This corresponds to a contact angle of $\pi/2$. Such colloids are strongly surface active; the interfacial energy is $4\pi a^2 \tilde{\sigma}$ for a colloid wherever it is placed in the two fluids, but placing it symmetrically across the interface between them reduces the area of fluid–fluid contact by πa^2. Hence a neutrally wet colloid is bound to the fluid–fluid interface by an energy $\pi a^2 \sigma$ with σ the fluid–fluid interfacial tension. In most circumstances this quantity vastly exceeds $k_B T$ and therefore colloidal adsorption to the interface is effectively irreversible. This is the basis of several technologies involving emulsions stabilized by solids, which are generally called 'Pickering emulsions' [31]. Studying new variants of these is a primary motivation for our development of an LB code to handle colloids in binary solvents. Our current implementation of the code is restricted to neutral wetting, but similar physics

Time Step: 001000   Time Step: 085000   Time Step: 200000
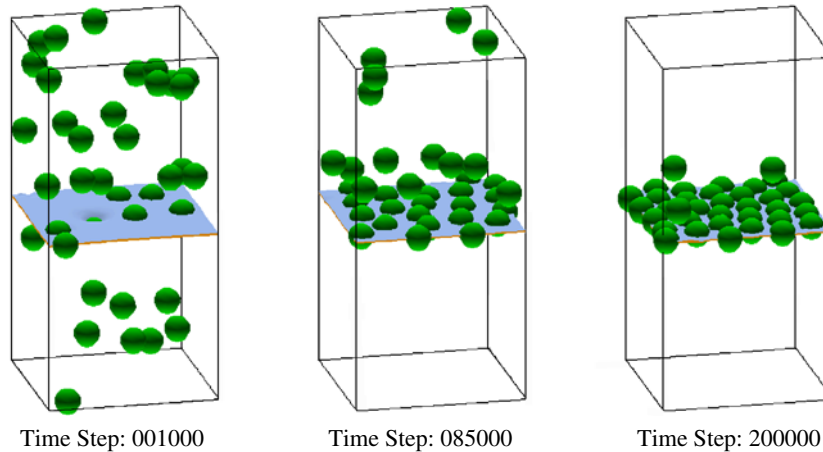
Figure 1. Snapshot configurations for the benchmark sedimentation problem with $Ca = 280$ and $Re = 0.02$.

should be seen for a range of angles around $\pi/2$. (The issue of how to vary the contact angle away from neutrality is understood in principle, and implemented but not yet tested in our colloid codes [12, 15].)

6. The benchmark problem and parameter selection

Before using a somewhat complicated algorithm (such as the LB one) to gain quantitative results for an equally complicated problem (such as colloids in binary solvents) it is important to complete a range of benchmark tests so as to give qualitative and quantitative insights into sources of systematic error. One must also develop a strategy for choosing simulation parameters to get close enough (given these sources of error) to some experimentally realizable system of interest. For binary fluids undergoing coarsening, extensive parameter testing is reported by Kendon *et al* [6], and we build on this wherever we can. But as far as colloids in binary solvents are concerned, the task is not yet finished: the results presented below are preliminary only.

To illustrate the principles involved, let us consider a simple geometry in which a suspension of colloids, at low volume fraction, sediments under gravity within a stratified pair of layers of two immiscible fluids of equal viscosity. Figure 1 gives a series of snapshots of this process; each shows the bottom half of a system with periodic boundary conditions. Thus, one fluid lies below the interface, another on top (and out of view above this is a second interface which restores periodicity in the vertical direction). The colloidal particles are initially placed at random, so some of them lie across the interface but most do not. They then fall under gravity and, for the parameters selected here, they all end up attached to the interface.

As is proper in a fluid mechanics problem, we address the physics of the situation by identifying some relevant dimensionless numbers. Indeed, these are particularly helpful in mapping between lattice units and the real world. The relevant numbers include the gravitational Reynolds and Peclet numbers [32]:

$$Re = \frac{v_{\text{sed}} a \rho}{\eta} \tag{9}$$

where $v_{\mathrm{sed}} = \Delta mg/6\pi\eta a$ is the sedimentation velocity, and

$$Pe = \frac{v_{\mathrm{sed}} a}{D} \tag{10}$$

where $D = k_{\mathrm{B}}T/6\pi\eta a$ is the diffusion constant of a colloid. Note that for terrestrial gravity g and typical density mismatch ($\Delta mg/mg \sim 0.1 - 1$, with $m = \rho 4\pi a^3/3$) one has Re typically of order $10^{-7}(a/a_\mu)^3$ and Pe of order $(a/a_\mu)^4$. Here we have introduced a convenient length scale $a_\mu \equiv 1\ \mu$m. Thus a colloid of, say, $a = 10\ \mu$m has negligibly small inertial effects (small Re) and also negligibly small diffusion (large Pe), when falling under gravity. For $a = 100$ nm, Re is still negligible but Pe is now small. Note that Pe may also be written as a/h, where $h = k_{\mathrm{B}}T/\Delta mg$ is the gravitational decay length for the colloidal concentration in barometric equilibrium.

The fluid–fluid interfacial tension σ introduces another dimensionless parameter, which we can write as a (gravitational) capillary number

$$Ca = \frac{v_0}{v_{\mathrm{sed}}}. \tag{11}$$

Here, v_0 is an intrinsic velocity scale σ/η which governs the dynamics of a disturbed interface between fluids [6]. Thus for large Ca, the interface can adapt easily to the arrival of particles but for small values the particles are likely to break through it as they sediment. Indeed, $Ca = 6\pi\sigma a/\Delta mg$, so, to within an order-unity factor, Ca also determines whether the force of attachment of a particle to the fluid–fluid interface is enough to hold it there against gravity. (At large Ca sedimenting particles should gather at the interface whereas for small Ca they will fall off it even if placed there gently.) Note that for typical binary fluid parameters, $Ca \simeq 3 \times 10^8 (a/a_\mu)^{-2}$. This large value (at $a \simeq a_\mu$) reflects the strength of the binding to fluid–fluid interfaces: particle sizes of order millimetres are needed before gravity will detach a neutrally wetting particle from, say, an oil–water interface. Note that a similar dimensionless quantity arises for the gravitational equilibrium of a pendant droplet of one fluid suspended against gravity in another by capillary forces; in this context, Ca^{-1} is usually called the Bond number.

Under conditions where sedimentation is strong and diffusion weak, we can set Pe to be infinite. This is in fact the default position in the LB method unless noise is explicitly added. (A way to do that is described in section 7.) More interesting is the role of the Reynolds and capillary numbers; we address these in turn.

As outlined above, for typical colloid parameters Re is extremely small. However, the LB method works by solving dynamically a discretization of the full hydrodynamic equations (6)–(8). This means in practice that Re can never be made fully realistic for colloids, as we now explain. A reasonable duration of a simulation is of order 10^4 (or perhaps 10^5) time steps; a reasonable lattice size of order 128^3 (achievable for repeat runs on a shared-memory machine) or perhaps 512^3 (for bespoke production runs on national supercomputer facilities). Note that in the LB method the computing resource per time step scales with reasonable linearity with system volume on these platforms; but linear volume scaling means that ten times as much computer resource buys little more than a factor two in length scale. If the phenomena of interest also take longer at larger scales (e.g. phase separation) the duration of the run must also be increased. For colloidal particles to move a significant distance in the run times achievable, their velocities v_{sed} must be of order 10^{-3} LU or more. However, the maximum safe viscosity is of order 1 LU; the colloid radius a is a few LU; and the density $\rho = 1$; therefore $Re = v_{\mathrm{sed}}\rho a/\eta$ must be set at about 10^{-2}, or else no colloids will move at all far during the course of the simulation. Re can be reduced from this by perhaps one or two orders of magnitude by reducing v_{sed}, but only at the cost of very long simulation times. Even this does not approach the tiny values of Re often encountered for real colloids.

However, in most problems, this does not matter: all Reynolds numbers below an appropriate (problem-dependent) threshold are already virtually equivalent. To see this, note that at low enough Re, the velocity field around a sedimenting particle is quasi-static; this Stokesian limit represents diffusive equilibrium of momentum. In principle one could use a variety of algorithms to solve the quasi-static problem. But whatever method was used, it would not be appropriate to do this to accuracy of one part in 10^7, when there are various per cent-level errors arising from other aspects of the code (such as the discretization of the colloid links in the bounce-back algorithm, and in the treatment of interfacial physics). Thus, the goal of the LB method should not be to simulate 'fully realistic' Re (of order 10^{-7}) but to use values of Re small enough that the resulting error, in the colloid and solvent dynamics, is no more than a few per cent. Such values represent a 'realistic' but not a 'fully realistic' simulation.

To explore this further, figure 2 shows contour plots for the magnitude of the difference $\Delta\mathbf{u}$ in the normalized velocity field $\mathbf{u} = \mathbf{v}/v_{\text{sed}}$ for various simulations of a single falling sphere (with periodic boundary conditions). Our 'reference' simulation has very small $Re = 3 \times 10^{-6}$. The flow is computed by working in the co-falling frame so that the velocity field converges to a time-independent limit on the lattice; this can be calculated with high accuracy. (Note that such a frame cannot be defined for a problem involving *more* than one sphere.) The resulting contour lines for $\mathbf{u}$ itself are shown in the first panel; difference plots for $Re = 0.008, 0.08$, and 0.8 are then presented. The latter simulations did not use the co-falling frame; the two higher values are in a range realizable for multi-sphere problems, using reasonable run times. We find that for $Re = 0.08$, $|\Delta\mathbf{u}| \leqslant 0.03$ through the main bulk of the fluid, with a slightly larger error close to the colloid. It is possible that this error is partly from the sudden shape changes of the particle as it crosses the lattice (rather than directly from the finite diffusivity of the momentum field). In any case, the deviation from the zero-Reynolds-number mobility for the falling sphere is about 2%. This suggests that $Re < 0.1$ represents an acceptably small value for most colloid simulations, and that somewhat larger values might even be acceptable (at least for qualitative exploration of parameter space prior to large-scale production runs; even at $Re = 0.8$ we find an error in v_{sed} of only about 5%). This concurs with the comments of Batchelor [33] that in flow past a sphere, all $Re < 1$ are practically equivalent. But note that Batchelor's remark addresses the nonlinear term in the Navier–Stokes system only, whereas our error stems also from the time derivative, since our flow is nonstationary in the lattice frame.

In conclusion, for most problems of interest, the level of systematic error caused by $Re < 0.1$, or more conservatively $Re \leqslant 0.05$, is probably acceptable. One possible exception lies in the study of velocity fluctuations in steady sedimentation at large length scales [34], whose subtle physics could amplify systematic errors that are negligible at the scale of a single colloid. (For lattice Boltzmann work on this topic see [35].)

We turn now to the role of the gravitational capillary number, Ca. We showed in figure 1 a series of snapshots for colloidal sedimentation in binary fluids with a stratified interface, at $Ca = 280$ and $Re = 0.02$. The simulated Ca is large, but still much smaller than would be typical for micron-scale colloids. Indeed, for typical interfacial tension σ between two fluids, to achieve this small a Ca in ordinary gravity would involve using millimetre sized particles— so heavy that the appropriate Re is actually *more* than the simulated one, unless the solvent viscosity is of order 1000 times larger than that of water. (Large particles in viscous solvents are studied in [34], and certainly do show interesting properties.) For viscosities similar to water, our parameters correspond to micron-scale particles, but in an ultrahigh gravitational field (of order 10^6 times the usual one, if maximal density mismatch is used).

However, this insistence on 'fully realistic' matching of dimensionless parameters between the simulations and laboratory is misguided. Judging from the sedimentation test reported
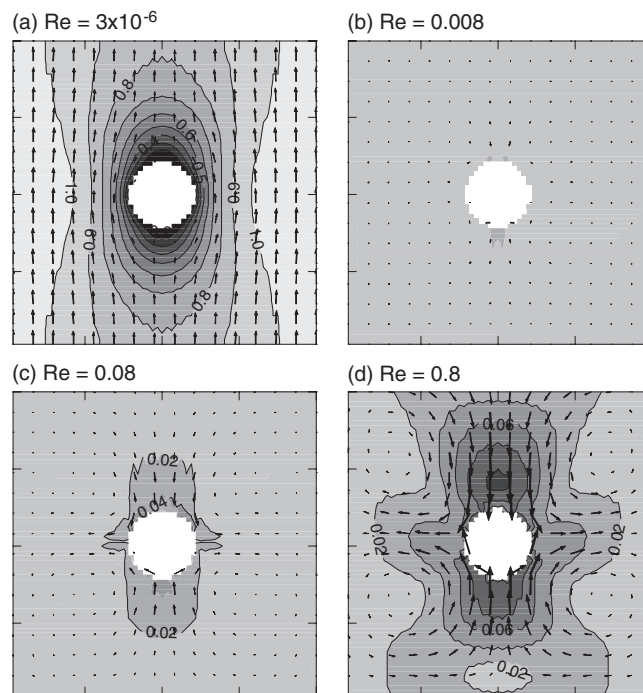
Figure 2. Contour plots showing the magnitude of the normalized velocity field **u** for a reference simulation at very low $Re = 3 \times 10^{-6}$ (top left); and velocity difference fields $|\Delta \mathbf{u}|$ for $Re = 0.008$, 0.08, and 0.8. These are for a sphere undergoing sedimentation in periodic boundary conditions. Reference case: contour interval 0.1. Other plots: contour interval 0.02. The arrows denote the velocity field/velocity difference field in each case (scaled similarly).

above, $Re = 0.02$ is small enough to represent any small value; numerical effort is only wasted by reducing it further. Similar remarks probably apply to Ca also. Thus, in principle we should increase σ by many orders of magnitude if we want to place Ca within the realistic experimental range for micron-scale colloids. But this would involve taking σ far beyond any values benchmarked in [6], and success appears unlikely, since the hugely enhanced interfacial forces would probably lead to numerical instability. However, it may well be that $Ca = 280$ is already large enough to reproduce, at least for our chosen benchmark problem, all the realistic colloid physics of the large-Ca limit. Perhaps, in fact, we can afford to reduce Ca by one or two orders of magnitude and still see the same physics. (Indeed, a similar run with Ca around 20 and $Re = 0.2$ gives no great differences from figure 1; data not shown.) Similar benchmarking issues will be important to a number of other problems involving colloids in binary solvents, and we leave it open to future study to find out how small a value of Ca is acceptably large, in this and related cases.

In summary, we can state that the LB method has a good prospect of achieving 'realistic' simulations in the sense of a proper relative hierarchy of importance among competing physical effects (capillary terms, diffusion, and inertia, as governed respectively by Ca, $1/Pe$, Re). But there is very little prospect of achieving 'fully realistic' simulations in the sense of actually resolving the several decades in time and length, between one effect and the next, that are present in typical laboratory experiments. The reason for this is primarily to do with computational resources, and it affects competing algorithms as well as the LB one. Currently

then, no simulation method for colloid hydrodynamics problems can be used reliably, unless it is supported by some physical insight into the nature of the problem being solved.

7. Adding noise to the LB approach

There are many physics problems involving colloid hydrodynamics where Brownian motion is important. The latter arises from the bombardment of the colloids by random forces from the surrounding fluid, and in the well-known simulation method of Brownian dynamics one simply adds a random force directly to each colloid [36]. However, this neglects many-body hydrodynamics which, among other things, induces nontrivial correlations between the noise forces acting on different colloidal particles. Over the years, various attempts have been made to address such correlations within Brownian dynamics, but to do this without ad hoc approximation means inverting a many-body mobility matrix at every time step. (That is indeed done in full Stokesian dynamics [29, 30].) This inversion has bad scaling with system size, and for large systems a local algorithm such as the LB one will generally prevail. The proper way to deal with noise in the LB method is to add random forces, not to the particles, but to the fluid itself and allow these to propagate, via the hydrodynamic fields, into the colloid sector.

At the continuum level, this amounts to adding a fluctuating stress to equation (8) [37]:

$$\Pi_{\alpha\beta} = g_\alpha v_\beta + p\delta_{\alpha\beta} - \eta_{\alpha\beta\gamma\delta}\nabla_\gamma v_\delta + s_{\alpha\beta}. \tag{12}$$

The fluctuating stress $s_{\alpha\beta}$ is a zero-mean Gaussian random variable whose variance, for a fluid at temperature T, is fixed by the fluctuation-dissipation theorem (FDT) to be $\langle s_{\alpha\beta}(\mathbf{x}, t)s_{\gamma\delta}(\mathbf{x}', t')\rangle = 2k_\mathrm{B}T\eta_{\alpha\beta\gamma\delta}\delta(\mathbf{x} - \mathbf{x}')\delta(t - t')$. One way forward, due to Ladd [9], then consists of adding a corresponding stochastic term to the microscopic stress tensor which enters the equilibrium distribution in equation (4). However, the numerical results from this are not accurate. This is because of the non-hydrodynamic degrees of freedom that (in order to maintain Galilean invariance and isotropy) are necessarily retained within the LB method alongside the hydrodynamic ones: the 'ghosts'. If the noise terms act only on the hydrodynamic modes, the ghosts continually drain thermal energy away so the hydrodynamics never reaches equilibrium.

An improved method is presented in [13], in which we promote the LBE, equation (4), into a discrete Langevin equation where the f_i are interpreted as instantaneous, fluctuating densities in phase space:

$$f_i(\mathbf{x} + \mathbf{c}_i, t + 1) = f_i(\mathbf{x}, t) + L_{ij}(f_i(\mathbf{x}, t) - f_i^0(\mathbf{x}, t)) + \xi_i(\mathbf{x}, t) \tag{13}$$

with noise terms $\xi_i(\mathbf{x}, t)$. To recover thermal equilibrium, the ξ_i *must* be linked, by a FDT, to the collisional dissipation. The derivation of the required FDT is quite subtle [13] and in practice requires a careful analysis of all sources of dissipation within the collision process, through a study of its eigenmodes. There are nontrivial correlations among the ξ_i at a given site and time step: specifically these must be interdependent in such a way as to exactly conserve ρ and g_α. For a general LB scheme in d dimensions containing n velocities (a 'DdQn model'), there are precisely n eigenvectors, corresponding to the n degrees of freedom contained in the f_i at a given site. A complete mode count then consists of one null eigenvector corresponding to the conserved density ρ; d null eigenvectors corresponding to the d conserved components of the momentum g_α; $\frac{1}{2}d(d + 1)$ eigenvectors corresponding to the deviatoric momentum flux; and the remaining $n - (1 + d + \frac{1}{2}d(d + 1))$ ghost mode eigenvectors.

We can formally set $\xi_i = \xi_i^H + \xi_i^G$, with H the hydrodynamic subspace and G its complement, the ghost subspace. Here ξ_i^H produces thermal fluctuations in the stress tensor,
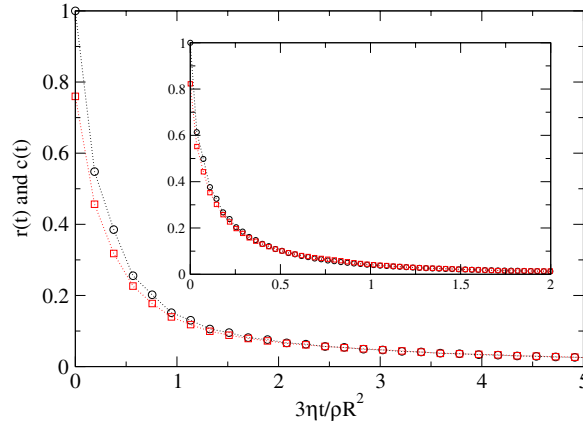
Figure 3. The velocity response (circles) and correlator (squares) for a colloid. Colloid radius $R = 2.3$; $\eta = 1/3$, $\mathcal{N} = 400$. D3Q15 lattice (BGK collision matrix). Inset: the angular velocity response (circles) and correlator (squares); $R = 3.71$, $\eta = 1/6$, $\mathcal{N} = 400$. Correlators represent data collected over 20 000 time steps, averaged over three Cartesian velocity components and ten colloidal particles.

and is the noise used by Ladd [9]. The remaining terms are ξ_i^G: these maintain thermal equilibrium for the ghosts. An explicit method for constructing appropriate noise is described in [13]. In practical simulations, we continue to set mass, length, and time units so that $\rho = 1$ on a unit lattice, and $c_s^2 = k_B T/\mu = 1/3$, where μ is the mass of one of our fictitious fluid particles. We then choose $\rho/\mu = \mathcal{N} \gg 1$ as the mean number of particles per lattice site. (If this inequality is not satisfied, the fluid ceases to be a continuum at the lattice scale.) Since the LB fluid is in fact an ideal gas, fluctuations then obey $\langle\delta\mathcal{N}^2\rangle = \mathcal{N} = c_s^2\rho/k_B T$.

The consistency of our approach can be assessed by measuring numerically the 'equilibration ratio' for fluctuating hydrodynamic quantities. This is the ratio of an actual variance (of, say, a Fourier amplitude of momentum) to the one required by thermal equilibrium at the temperature T chosen for the simulation. As shown in [13] our method gives equilibration ratios within a few per cent of unity, whereas omitting ξ_i^G gives discrepancies of order 30%. Note that in the method of [28], the dissipative coupling between colloids (treated as pointlike) and fluid is accompanied by noise terms which may swamp those arising directly from the fluid itself. This could resolve the problem with equilibration—or it could merely mask it. In particular, correlations between different colloidal particles must depend on the proper transfer of noise forces through the solvent; so one should carefully check for errors in these quantities.

Figure 3 shows the impulse response function $r(t) = \langle v(t)\rangle/v(0)$, and velocity autocorrelator $c(t) = M\langle v(0)v(t)\rangle/kT$ for a Cartesian component $v(t)$ of the velocity of a colloidal particle of mass M. The colloid is suspended in a quiescent fluctuating fluid of equal density, in three dimensions on a D3Q15 lattice [13]. This is a parameter-free comparison; FDT demands $r(t) = c(t)$. We find excellent agreement for time intervals beyond a few time steps; in fact $c(t + \Delta) = r(t)$ to high accuracy even at short times (not shown), where the best offset Δ depends slightly on parameters, but is about 0.5. We attribute the offset to imperfect resolution of a rapid sound-mediated decay at very short times. (This is singular for an incompressible fluid, with $c(0+) = 2/3$.) The offset causes a slight deficit, at most a few per cent, in the colloidal self-diffusion constant $D = \int_0^\infty \langle v(0)v(t)\rangle\,dt$ from its FDT value. There is a bigger deficit in the equal time correlator $c(0)$, but even this becomes accurate for larger colloids, where the fast decay is better resolved. (For practical colloid simulation,

D matters rather than *c*(0).) The inset of figure 3 shows corresponding data for rotational degrees of freedom; agreement is again excellent. We conclude that, although some issues remain concerning the choice of optimal parameters and run times for problems involving Brownian motion of colloids, the fluctuating LBE presented in [13] represents a very promising way forward for simulating such problems.

8. Conclusion

In this paper we have outlined recent progress in the study of colloid hydrodynamics using lattice Boltzmann methods. We have made significant progress in the description of colloids in binary solvents and also for colloids in thermal solvents undergoing Brownian motion. Both of these aspects of the code appear to behave sensibly, and in the near future we hope to perfect these tools and apply them to a variety of interesting scientific questions beyond the simple benchmark problem of figure 1.

Acknowledgments

References

[1] Succi S 2001 *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond* (Oxford: Clarendon)
[2] Amati G, Succi S and Priva R 1997 *Int. J. Mod. Phys.* C **8** 869
[3] Lockard D P, Luo L S, Milder S D and Singer B A 2002 *J. Stat. Phys.* **107** 423
[4] Discrete modelling and simulation of fluid dynamics *Phil. Trans. R. Soc.* A **360**
   Coveney P V and Succi S (ed) 2002 *Proc. 9th Int. Conf. on Discrete Simulation of Fluid Dynamics*; *J. Stat. Phys.* **107** 1–592
[5] Manz B, Gladden L F and Warren P B 1999 *AICHE J.* **45** 1845
[6] Kendon V M, Cates M E, Pagonabarraga I, Desplat J C and Bladon P 2001 *J. Fluid Mech.* **440** 147
   Kendon V M, Desplat J C, Bladon P and Cates M E 1999 *Phys. Rev. Lett.* **83** 576
[7] Pagonabarraga I, Desplat J-C, Wagner A J and Cates M E 2001 *New J. Phys.* **3** (art 9)
[8] Nekovee M, Coveney P V, Chen H D and Boghosian B M 2000 *Phys. Rev.* E **62** 8282
[9] Ladd A J C 1994 *J. Fluid Mech.* **271** 285
   Ladd A J C 1994 *J. Fluid Mech.* **271** 311
[10] Swift M R, Osborn W R and Yeomans J M 1995 *Phys. Rev. Lett.* **75** 830
[11] Love P J, Nekovee M, Coveney P V, Chin J, Gonzalez-Segredo N and Martin J M R 2003 *Comput. Phys. Commun.* **153** 340
[12] Stratford K *et al* 2004 in preparation
[13] Adhikari R, Cates M E, Stratford K and Wagner A 2004 *Preprint* cond-mat/0402598
[14] Wagner A J and Pagonabarraga I 2002 *J. Stat. Phys.* **107** 521
[15] Desplat J-C, Pagonabarraga I and Bladon P 2001 *Comput. Phys. Commun.* **134** 273
[16] Wagner A J 2001 *Preprint* cond-mat/0105067
[17] Wagner A J 2003 *Int J. Mod. Phys.* B **17** 193
   Adhikari R 2004 work in progress
[18] Pagonabarraga I, Wagner A J and Cates M E 2002 *J. Stat. Phys.* **107** 39
[19] Wagner A J and Cates M E 2001 *Europhys. Lett.* **56** 556
[20] Wagner A J, Cates M E and Vollmer J 2003 *Proc 3rd Int. Conf. on Computational Heat and Mass Transfer (Banff, May 2003)* ed A A Mohamad, p 40
   Wagner A J, Cates M E and Vollmer J 2004 in preparation
[21] Wagner A J, Wilson L M and Cates M E 2003 *Phys. Rev.* E **68** 045301
[22] Xi H W and Duncan C 1999 *Phys. Rev.* E **59** 3022
[23] Stansell P *et al* 2004 work in progress
[24] Eggers J 1997 *Rev. Mod. Phys.* **69** 865

[25] Aidun C K, Lu Y and Ding E-J 1998 *J. Fluid Mech.* **373** 287

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

[27] Nguyen N-Q and Ladd A J C 2002 *Phys. Rev.* E **66** 046708

[28] Ahlrichs P and Dunweg B 1999 *J. Chem. Phys.* **111** 8225

Ahlrichs P, Everaers R and Dunweg B 2001 *Phys. Rev.* E **64** 040501

[29] Ball R C and Melrose J R 1997 *Physica* A **247** 444

Silbert L E, Melrose J R and Ball R C 1999 *Mol. Phys.* **96** 1667

[30] Phung T N, Brady J F and Bossis G 1996 *J. Fluid Mech.* **313** 181

Foss D R and Brady J F 2000 *J. Fluid Mech.* **407** 167

[31] Aveyard R, Binks B P and Clint J H 2003 *Adv. Colloid Interface Sci.* **100–102** 503

[32] Chaikin P 2000 *Soft and Fragile Matter, Nonequilibrium Dynamics, Metastability and Flow* ed M E Cates and
M R Evans (Bristol: Institute of Physics Publishing)

[33] Batchelor G K 1967 *An Introduction to Fluid Dynamics* (Cambridge: Cambridge University Press) pp 230–46

[34] Segrè P N, Herbolzheimer E and Chaikin P M 1997 *Phys. Rev. Lett.* **79** 2574

[35] Ladd A J C 2002 *Phys. Rev. Lett.* **88** 048301

Ladd A J C 1997 *Phys. Fluids* **9** 491

Ladd A J C 1996 *Phys. Rev. Lett.* **76** 1392

[36] see, e.g. Foss D R and Brady J F 2000 *J. Rheol.* **44** 629

[37] Lifshitz E M and Pitaevski L P 1981 *Physical Kinetics* (Oxford: Pergamon) section 19