

Coupled lattice Boltzmann method and discrete element modelling of particle transport in turbulent fluid flows: Computational issues

Y. T. Feng^{*,†}, K. Han and D. R. J. Owen

Civil and Computational Engineering Centre, School of Engineering, University of Wales Swansea, SA2 8PP, U.K.

SUMMARY

This paper presents essential numerical procedures in the context of the coupled lattice Boltzmann (LB) and discrete element (DE) solution strategy for the simulation of particle transport in turbulent fluid flows. Key computational issues involved are (1) the standard LB formulation for the solution of incompressible fluid flows, (2) the incorporation of large eddy simulation (LES)-based turbulence models in the LB equations for turbulent flows, (3) the computation of hydrodynamic interaction forces of the fluid and moving particles; and (4) the DE modelling of the interaction between solid particles. A complete list is provided for the conversion of relevant physical variables to lattice units to facilitate the understanding and implementation of the coupled methodology. Additional contributions made in this work include the application of the Smagorinsky turbulence model to moving particles and the proposal of a subcycling time integration scheme for the DE modelling to ensure an overall stable solution. A particle transport problem comprising 70 large particles and high Reynolds number (around 56 000) is provided to demonstrate the capability of the presented coupling strategy. Copyright © 2007 John Wiley & Sons, Ltd.

Received 14 August 2006; Revised 4 May 2007; Accepted 4 May 2007

KEY WORDS: lattice Boltzmann method; discrete element method; fluid–particle interaction; immersed boundary scheme; Smagorinsky turbulence model

1. INTRODUCTION

Particle transport in fluid flows has a wide range of applications in science and engineering. The development of an effective numerical modelling framework for this type of problem is, however, very challenging due to the inherent complexity in the problem.

The numerical treatment of particle transport problems may crucially depend on the size of particles in relation to domain/mesh size. For small particle sizes, as occur in fluidized beds for example, effective methods have been developed based on background fluid grids in which the

*Correspondence to: Y. T. Feng, Civil and Computational Engineering Centre, School of Engineering, University of Wales Swansea, SA2 8PP, U.K.

†E-mail: y.feng@swansea.ac.uk

fluid-phase hydrodynamics is described by the volume-averaged (compressible) Navier–Stokes equations; empirical-based fluid–particle forces are applied to particles residing within a particular grid cell, which in turn determines the local porosity of the cell to be used in the fluid-phase computation [1, 2]. However, for problems in which particle sizes are large and extend over more than one grid/element size—given the need for a minimum grid/element size to resolve the fluid problem—alternative solution strategies must be adopted.

Over the last two decades, substantial effort has been made in the development of various numerical techniques to model fluid–particle interactions along with significant advances in effective computational solution procedures for general Navier–Stokes equations. One notable example is the work of Tezduyar and co-workers. They develop a general space–time finite element method for moving boundary and interface problems (see [3] for a review), which is also suitable for fluid–particle systems. Such a method seeks to achieve high accuracy in modelling the interaction between fluid and particles. It is therefore a valuable predictive tool for establishing the relationship between particle–fluid interaction force and the characteristics of fluid flow particles. Although it has been employed to simulate problems involving many particles, it may not be sufficiently effective for modelling practical problems with a large number of particles, particularly in 3D cases. Due to its Lagrangian nature, the main computational obstacle with this method is the need to continuously create new geometrically adapted meshes to circumvent severe mesh distortion, which is very computationally intensive, particularly in 3D situations. An even greater challenge is the generation of a valid mesh for (locally) dense particle flows where particle ‘kissing’ or collision may be a dominant physical phenomenon. Also note that many simulated flow problems are laminar with low Reynolds numbers, while practical particle transport problems are mainly of a turbulent nature.

In addition to conventional finite element and finite volume-based fluid solvers, the lattice Boltzmann (LB) method has emerged as a powerful alternative tool for the solution of fluid flows. Originally proposed in the 1980s as an improvement of general lattice gas modelling techniques, the LB method can be viewed as a ‘micro-particle’-based explicit time-stepping procedure for the solution of incompressible fluid flows, using a fixed regular grid. Detailed theoretical analysis has placed the LB method on a solid foundation, and a large number of numerical validations have also been conducted to assess its accuracy. See [4] and the references therein for a comprehensive review. From a computational point of view, the Eulerian and explicit natures of the LB method make it particularly suitable for fluid–particle problems, and large sized particles can be readily accommodated within its framework. With Ladd’s modification [5] to the so-called bounce-back rule to enforce the essential ‘no-slip’ boundary condition between the fluid and moving particles, the LB method has been successfully applied to practical problems, especially particle suspensions. A distinct advantage of the LB method over conventional methods lies in its capability of modelling problems with complex interior fluid domains and with a large number of particles.

The standard LB formulation is developed for laminar flows at low Reynolds numbers (typically around 100 or less). Many engineering problems nevertheless exhibit turbulence behaviour. Recently, limited attempts have been made to incorporate some existing turbulence models into the LB framework. The conventional large eddy simulation (LES) approach with a one-parameter Smagorinsky subgrid model [6] appears to be the simplest to apply; the more supplicated two-parameter κ – ε turbulence model [7] has also been considered. The current development in this aspect can be found, for instance, in [7–10]. It is highlighted that most reported work deals with problems with stationary particles/obstacles.

In many LB simulations of particle–fluid problems, the inter-particle interactions are either ignored or simply treated. However, more accurate resolutions are required for particle interactions in most practical applications. Employing the discrete element method (DEM), another numerical solution strategy developed particularly for discrete and particulate systems, to more accurately account for particle interactions has led naturally to a combined LB–DE solution procedure. The explicit nature of both LB and DE methods makes them an ideal partner from the computation and implementation points of view. Such a coupled methodology was first proposed by Cook *et al.* [11] in simulating particle–fluid systems dominated by particle–fluid and particle–particle interactions, and proved to be a powerful predictive tool for gaining fundamental insights into many poorly understood physical phenomena in particle–fluid systems.

This paper attempts to present essential numerical techniques in the context of the combined LB–DE solution strategy for a successful simulation of particle–fluid systems with a large number of particles and high Reynolds numbers [12]. Key computational issues involved comprise the standard LB formulation for fluid flows, the incorporation of turbulence models for high Reynolds number cases, the interaction description between fluid flows and moving particles, and the DE modelling of the interaction of solid particles. The majority of the techniques to be presented are well documented in the literature. However, there exists a rich body of literature on the topic, and so it is not trivial to identify appropriate techniques for the problems under consideration. Meanwhile, the differences between two or more available options for either modelling or implementation are often not clear. It is therefore felt that a relatively comprehensive account of proven working techniques and their associated defects (if any) for the modelling of particle–fluid systems may serve as a useful reference for further development.

In addition, various LB formulations in the literature are described in either physical or lattice units, which may cause confusion to less experienced researchers in the field. A complete list is thus provided for the conversion between these two unit systems to facilitate the understanding and implementation of the methodology. Other contributions made in this work include the application of the one-parameter Smagorinsky subgrid turbulence model to moving particles and the proposal of a subcycling time integration scheme for the DEM part to ensure an overall stable LB–DE solution. A particle transport problem involving 70 large particles and high Reynolds number (around 56 000) is provided to demonstrate the capability of the presented coupling strategy.

2. STANDARD LB FORMULATIONS FOR 2D INCOMPRESSIBLE FLUID FLOWS

This section describes standard LB formulations and associated modelling and implementation issues for the solution of 2D incompressible fluid flows.

2.1. D2Q9 formulations

Consider a 2D incompressible fluid flow, with density ρ and kinematic viscosity ν , in a rectangular domain \mathbb{D} . The fluid domain is first divided into a regular grid, or lattice, with spacing h in both x and y directions, as shown in Figure 1(a). The fluid phase is then represented as a group of fluid ‘particles’ residing at lattice nodes that move to the neighbouring nodes along a fixed set of discrete directions with given discrete velocities at discrete times. In the widely used 2D LB discretization scheme, the so-called D2Q9 model [13], the fluid particles at each node are allowed to move to its eight immediate neighbours with eight different velocities \mathbf{e}_i , ($i = 1, \dots, 8$), as

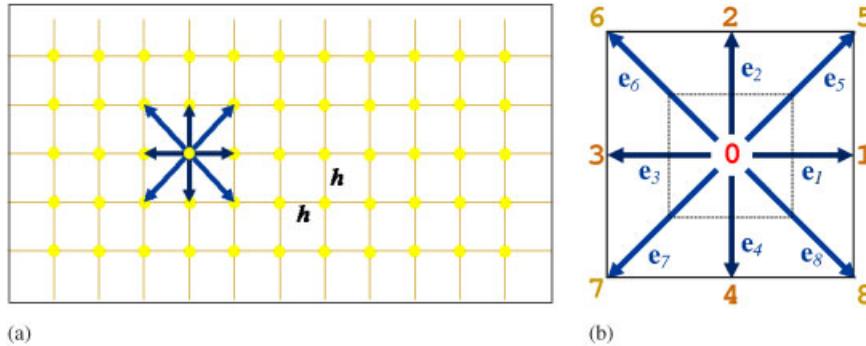


Figure 1. LB discretization and D2Q9 model: (a) a standard LB lattice and (b) D2Q9 model.

shown in Figure 1(b). A proportion of the particles can rest at the node, which is equivalent to moving with a zero velocity \mathbf{e}_0 . Using the numbering system shown in Figure 1(b), these nine discrete velocity vectors in total are defined as

$$\begin{aligned} \mathbf{e}_0 &= (0, 0) \\ \mathbf{e}_i &= C \left(\cos \frac{\pi(i-1)}{2}, \sin \frac{\pi(i-1)}{2} \right) \quad (i = 1, \dots, 4) \\ \mathbf{e}_i &= C \left(\cos \frac{\pi(2i-9)}{4}, \sin \frac{\pi(2i-9)}{4} \right) \quad (i = 5, \dots, 8) \end{aligned} \tag{1}$$

in which C is termed the lattice speed and given by

$$C = h/\Delta t$$

with Δt the discrete time step. The primary variables in the LB formulation are the so-called *fluid density distribution functions*, f_i , each relating to the probable number of fluid particles moving with velocity \mathbf{e}_i along the i th direction at each node. There are thus nine fluid density distribution functions, f_i ($i = 0, \dots, 8$), associated with each node in the D2Q9 model. At each time step, the fluid particles residing at the original node move to the nearest nodes along the given directions where particle collisions and new distributions take place. The central issue to the LB formulation is to define the rule that governs the movement of fluid particles in the lattice at discrete time instances. Following the popular linearized single-relaxation-time BGK formulation [4], the evolution of the density distribution functions at each time step is given by

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{\tau} [f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)] \quad (i = 0, \dots, 8) \tag{2}$$

where for any grid node \mathbf{x} , $\mathbf{x} + \mathbf{e}_i \Delta t$ is its nearest node along the direction i ; τ is a non-dimensional parameter termed the relaxation time; and f_i^{eq} are the equilibrium distribution functions

defined as

$$\begin{aligned}
 f_0^{\text{eq}} &= w_0 \rho \left(1 - \frac{3}{2C^2} \mathbf{v} \cdot \mathbf{v} \right) \\
 f_i^{\text{eq}} &= w_i \rho \left(1 + \frac{3}{C^2} \mathbf{e}_i \cdot \mathbf{v} + \frac{9}{2C^4} (\mathbf{e}_i \cdot \mathbf{v})^2 - \frac{3}{2C^2} \mathbf{v} \cdot \mathbf{v} \right) \quad (i = 1, \dots, 8)
 \end{aligned}
 \tag{3}$$

in which w_i are the weighting factors:

$$w_0 = \frac{4}{9}, \quad w_{1,2,3,4} = \frac{1}{9}, \quad w_{5,6,7,8} = \frac{1}{36}
 \tag{4}$$

In addition, the right-hand side of (2) is often denoted by $f_i(\mathbf{x}, t_+)$ and termed the post-collision distribution.

The LB discrete time evolution equation (2) ensures conservations of total mass and total momentum of the fluid particles at each lattice node. The computation comprises essentially two phases: *collision* and *streaming*. The collision operation computes the right-hand side of Equation (2) that only involves the variables associated with each node \mathbf{x} , and therefore is a local operation. The streaming phase then explicitly propagates the updated distribution functions at each node to its neighbours $\mathbf{x} + \mathbf{e}_i \Delta t$, where clearly no computations are required and only data exchange between neighbouring nodes is necessary. These features, together with the explicit time-stepping nature and the use of a regular grid, make LB computationally efficient, simple to implement and natural to parallelize.

The macroscopic fluid variables, density ρ and velocity \mathbf{v} , can be recovered from the moments of the distribution functions as follows:

$$\rho = \sum_{i=0}^8 f_i, \quad \rho \mathbf{v} = \sum_{i=1}^8 f_i \mathbf{e}_i
 \tag{5}$$

while the fluid pressure field p is determined by the following equation of state:

$$p = C_s^2 \rho
 \tag{6}$$

where C_s is termed the fluid speed of sound and is related to the lattice speed C by

$$C_s = C / \sqrt{3}
 \tag{7}$$

The kinematic viscosity of the fluid, ν , is, however, not directly used in the LB model but implicitly determined by the discretization and numerical parameters h , Δt and τ as

$$\nu = \frac{1}{3} \left(\tau - \frac{1}{2} \right) \frac{h^2}{\Delta t} = \frac{1}{3} \left(\tau - \frac{1}{2} \right) Ch
 \tag{8}$$

2.2. Convergence, stability and model parameter selection

It can be proved that the LB equation (2) recovers the incompressible Navier–Stokes equations to second order in both space and time [4, 14], which is the theoretical foundation for the success of the LB method for modelling general fluid flow problems. However, since the LB formulation (2) is obtained by the linearized expansion of the original kinetic theory-based LB equation,

the resulting macroscopic variables converge to the solution of the incompressible Navier–Stokes equations with order Ma^2 , where Ma is termed the ‘computational’ Mach number and defined by

$$Ma = \frac{v_{\max}}{C} \quad (9)$$

where v_{\max} is the maximum simulated velocity in the flow. It is therefore required that

$$Ma \ll 1 \quad (10)$$

i.e. the lattice speed C should be sufficiently larger than the maximum fluid velocity to ensure a reasonably accurate solution. In practice, Ma should be, at least, smaller than 0.1.

On the other hand, the pressure state of Equation (6) reveals that the incompressibility of the fluid is not exactly enforced but satisfied approximately in the LB formulation. In other words, the LB model can be viewed as a ‘penalty-based’ method that allows a limited degree of compressibility to occur where the fluid speed of sound C_s or the lattice speed C acts as the penalty value. Again, a larger value of C will achieve a better approximation to the incompressibility condition.

There are three model parameters in the LB formulation: the lattice spacing h , the relaxation time τ and the time step Δt . Relation (8) indicates that the selection of these three parameters cannot be made independently but has to be related in order to achieve a correct fluid viscosity. More importantly, the parameters should be chosen such that a desired solution accuracy can be achieved with reasonable computational costs.

The lattice spacing h mainly determines the spatial discretization error of the LB model and its value has a direct effect on model size and thus computational cost of a simulation. The time step Δt will affect the temporal discretization error and the total number of steps required for a simulation. In general, the solution error may be expressed as $O(h^2) + O(Ma^2)$; thus to obtain an effective LB algorithm, the compressibility error (measured by Ma) has to be in balance with the resolution error (controlled by h).

The relaxation time τ is the most important model parameter in LB computations. It characterizes the time-scale behaviour of fluid particle collisions and determines the lattice fluid viscosity \bar{v} in the lattice scale:

$$\bar{v} = \frac{1}{3}(\tau - \frac{1}{2}) \quad (11)$$

The (true) fluid viscosity and the lattice viscosity have the following relation:

$$\nu_s = \bar{v} \frac{h^2}{\Delta t} = \bar{v} C^2 \Delta t \quad (12)$$

It is obvious from both (8) and (11) that τ has to be

$$\tau > \frac{1}{2}$$

τ is also largely responsible for the numerical stability of LB modelling. Generally speaking, a larger value represents a more viscous fluid and the LB simulation is more stable, whilst a smaller value corresponds to a less viscous fluid and the LB formulation is more prone to numerical instability, particularly when τ is approaching 0.5. This can also be understood from the fact that the limitation to the value of τ is imposed by the explicit feature of the LB formulation. It may be of both theoretical and practical importance if the critical value of τ , up to which a stable solution can be achieved, is known. Theoretically, this value satisfies an equation but it is highly non-linear

and strongly dependent on the actual flow pattern. As a result, it is impractical to attain unless for a very simple flow case [15, 16].

From a computational point of view, it is more convenient to choose h and τ as two independent parameters and Δt as the derived parameter from (8) as

$$\Delta t = \left(\tau - \frac{1}{2} \right) \frac{h^2}{3\nu} = h^2 \frac{\bar{\nu}}{\nu} \quad (13)$$

Since there is no *a priori* error estimation available to determine the appropriate values of h and τ for a fluid flow problem with given fluid viscosity ν , the following procedure may have to be adopted to obtain a convergent solution: (1) Give a spacing h mainly from the consideration of computational costs, choose a value of τ from the stability consideration. The time step Δt is then determined from (13) and the corresponding lattice speed C can now be calculated. (2) Perform the LB simulation following (2). If a stable solution is achieved, obtain the maximum fluid velocity and check the Mach number Ma . If Ma is sufficiently small, a reasonably accurate solution may have been achieved. Otherwise, h needs to be reduced. If the numerical solution is not stable, a larger τ may be needed. Repeat the above procedure until a satisfactory result is obtained.

Equation (13) shows clearly that for a given ν , and with τ (i.e. $\bar{\nu}$) unchanged, reducing h by half will result in the decrease of Δt by a quarter, thereby doubling the lattice speed C (i.e. reducing Ma) and leading to a better solution accuracy, but computer costs will quadruple at the same time. On the other hand, with the lattice speed C unchanged, reducing h will increase the value of τ as shown in (8), thereby enhancing the stability of the LB simulation.

The real challenge is to model fluid flows with a small viscosity, such as water and air. In these cases either h needs to be very small or τ has to be very close to 0.5, as is evident from (8). The former option may give rise to a prohibitively large-scale model, while the latter may suffer from numerical instability. The underlying problem is that the standard LB method is formulated for laminar flows, which is not applicable for problems with small kinematic viscosity as they are associated with large Reynolds numbers, or being turbulent in nature. Inevitably the LB formulation will diverge in these situations, unless the spatial resolution is comparable to the small scale of turbulence to be modelled. Therefore, the incorporation of a turbulence model into the LB formulation is essential for the LB modelling of many practical applications, which will be elaborated upon in Section 3.

The main disadvantage of the standard LB approach is its inflexibility to employ irregular or non-uniform meshes. Though some effort has been made to extend the LB formulation to non-uniform grids [17] or to apply local grid refinement [18], the implications caused by the extension might sometimes offset the benefits the resulting scheme can offer.

2.3. Lattice coordinates

The LB formulation in the preceding subsections is introduced in the physical unit system, in which all the relevant physical variables are in their standard units. In the literature, the LB formulation is often described in lattice units where all the variables can be *non-dimensionalized*, thereby offering an implementation advantage. In the present work, the LB method is also implemented using the lattice units with necessary conversions between the physical and lattice systems. In order to facilitate the implementation and gain a better understanding of the LB method, a complete list is given in Appendix A for the conversion of the variables used in the LB model between the two unit systems, where an overbar ‘ $\bar{}$ ’ is used to indicate a variable in the lattice units.

Note that the density distribution functions f_i and the relaxation time τ are already non-dimensional and thus remain the same in both systems. In the lattice system, the spacing \bar{h} and the time step $\bar{\Delta t}$ are both unity, leading to a unit lattice speed $\bar{C} = 1$. The lattice density is defined by

$$\bar{\rho} = \rho / \rho_0$$

where ρ_0 is the true fluid density. Theoretically ρ should be equal to ρ_0 , but they are slightly different because the incompressibility is not satisfied exactly. Therefore, $\bar{\rho}$ will be close to unity. Also in the lattice system, the indices of the grid nodes become their coordinates $\bar{\mathbf{x}}$, and time instances \bar{t} are now integer numbers. Then, the LB equations can be converted to lattice units by replacing all the variables with their counterparts in the lattice system. For instance, Equation (2) becomes

$$f_i(\bar{\mathbf{x}} + \bar{\mathbf{e}}_i, \bar{t} + 1) = f_i(\bar{\mathbf{x}}, \bar{t}) - \frac{1}{\tau} [f_i(\bar{\mathbf{x}}, \bar{t}) - f_i^{\text{eq}}(\bar{\mathbf{x}}, \bar{t})] \quad (i = 0, \dots, 8) \tag{14}$$

and the equilibrium distribution functions f_i^{eq} take the form of

$$\begin{aligned} f_0^{\text{eq}} &= w_0 \bar{\rho} (1 - \frac{3}{2} \bar{\mathbf{v}} \cdot \bar{\mathbf{v}}) \\ f_i^{\text{eq}} &= w_i \bar{\rho} (1 + 3 \bar{\mathbf{e}}_i \cdot \bar{\mathbf{v}} + \frac{9}{2} (\bar{\mathbf{e}}_i \cdot \bar{\mathbf{v}})^2 - \frac{3}{2} \bar{\mathbf{v}} \cdot \bar{\mathbf{v}}) \quad (i = 1, \dots, 8) \end{aligned} \tag{15}$$

Clearly, both (14) and (15) are simpler and computationally more efficient than (2) and (3).

2.4. Boundary conditions

As velocity and pressure are not primary variables in the LB formulation, the normal pressure, velocity and mixed boundary conditions cannot be imposed directly, but alternative conditions in terms of the distribution functions need to be derived. Three basic boundary conditions, including ‘no-slip’, pressure and velocity, are discussed and the equivalent LB conditions are given mainly following the work of Zou and He [19]. Other boundary conditions, such as periodic and stress-free conditions, can be found, for instance, in [20, 21]. The discussions will refer to a simple case as shown in Figure 2(a), where both the top and bottom sides of the domain are assumed to be solid walls and the left and right are taken as inlet and outlet flow boundaries.

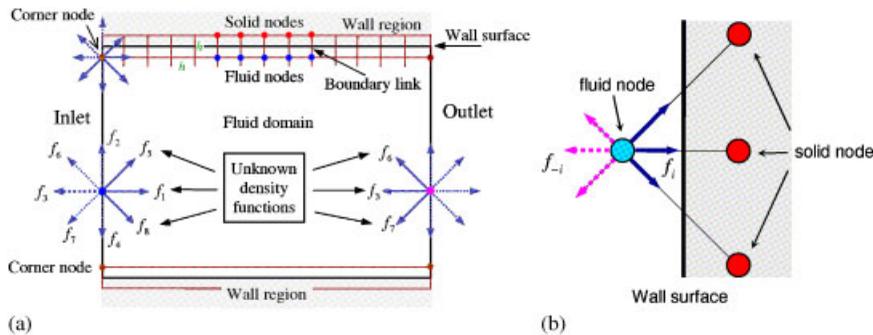


Figure 2. LB boundary conditions and bounce-back rule: (a) definition and (b) bounce-back rule.

2.4.1. *'No-slip' boundary conditions.* The 'no-slip' boundary condition at the interface of the fluid and stationary solid wall can be easily imposed by the so-called bounce-back rule [5]. Suppose that a solid wall is defined by a set of grid nodes, as shown in Figure 2(a). The nodes lying within the wall region are called *solid* nodes and the nodes in the fluid region are *fluid* nodes. If i is a link (or direction) between a fluid node and a solid node, the bounce-back rule requires that the incoming fluid particle from the fluid node be reflected back to the node it comes from, i.e.

$$f_{-i}(\mathbf{x}, t + 1) = f_i(\mathbf{x}, t_+) \tag{16}$$

where $-i$ denotes the opposite direction of i . This simple rule ensures that no tangential velocity exists along the fluid–wall interface; therefore, a 'no-slip' condition is imposed. The bounce-back rule is illustrated in Figure 2(b). Note that the wall boundary is assumed to be situated halfway between a fluid and solid node so as to achieve second-order accuracy. Otherwise, the accuracy will be first order. This bounce-back rule works reasonably well for stationary walls and can also be extended to any shaped wall or obstacle (including stationary particles) in fluid flows. It is this feature that makes the LB method particularly suitable for the modelling of problems with very complex boundaries, such as flow through porous media problems.

The bounce-back rule can be accomplished in two slightly different ways that have an implication on how the LB method is implemented. In the one-step scheme, the bounce-back rule (16) is enforced in the streaming phase of the current time step for the fluid nodes connected with a solid node, and all the solid nodes can be excluded from the computation; whereas in the two-step scheme, condition (16) is accomplished within two time steps. In the streaming phase of the current time step, f_i at the fluid node concerned is first propagated to the related solid node. Then in the collision phase of the next step, f_i is assigned directly to f_{-i} at the solid node, i.e. $f_i^{\text{eq}} = 0$ and no equilibrium calculation of (3) is performed at the solid node. Thus, f_{-i} is propagated back to the fluid node in the streaming phase. These two schemes have both advantages and disadvantages and the two-step scheme is used in the present work.

2.4.2. *Pressure boundary conditions.* Consider the rectangular domain where the pressure boundary conditions are applied to both inlet (left) and outlet (right) boundaries as shown in Figure 2. The pressure boundary conditions are imposed by specifying the fluid densities at the flow boundaries. Suppose that the inlet and outlet densities are ρ_{in} and ρ_{out} , respectively. The key to obtaining the equivalent LB conditions is to determine all the density functions at the boundary nodes, which requires the assumption of $u_y = 0$ at the boundaries and then the solution of u_x .

For an inlet boundary node as shown in the figure, the density functions f_i ($i = 2, 3, 4, 6, 7$) are known after streaming, but f_1, f_5 and f_8 have to be determined. Based on Equation (5) the following relations hold:

$$f_1 + f_5 + f_8 = \rho_{\text{in}} - (f_0 + f_2 + f_3 + f_4 + f_6 + f_7) \tag{17}$$

$$f_1 + f_5 + f_8 = \rho_{\text{in}} u_x + (f_3 + f_6 + f_7) \tag{18}$$

$$f_5 - f_8 = -f_2 + f_4 - f_6 + f_7 \tag{19}$$

Further using the bounce-back rule for the non-equilibrium part of the particle distribution normal to the inlet, it follows that

$$f_1 - f_1^{\text{eq}} = f_3 - f_3^{\text{eq}} \tag{20}$$

Then the solution is obtained from (17)–(20) as:

$$\begin{aligned}
 u_x &= 1 - \frac{f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)}{\rho_{\text{in}}} \\
 f_1 &= f_3 + \frac{2}{3}\rho_{\text{in}}u_x \\
 f_5 &= f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{\text{in}}u_x \\
 f_8 &= f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{\text{in}}u_x
 \end{aligned} \tag{21}$$

Similarly, the unknown density functions f_3 , f_6 and f_7 for the outlet nodes can be computed by

$$\begin{aligned}
 u_x &= -1 + \frac{f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)}{\rho_{\text{out}}} \\
 f_3 &= f_1 - \frac{2}{3}\rho_{\text{out}}u_x \\
 f_6 &= f_8 - \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_{\text{out}}u_x \\
 f_7 &= f_5 + \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_{\text{out}}u_x
 \end{aligned} \tag{22}$$

For the corner nodes, the above formulations are not applicable because of the singularity at the nodes. Consider the top-left corner node in Figure 2(a) as an example. After streaming, f_2 , f_3 and f_6 are known; ρ is specified, and $u_x = u_y = 0$. Five unknown density functions f_1 , f_4 , f_5 , f_7 , f_8 need to be determined. Again applying the bounce-back rule to the non-equilibrium part of the distribution normal to the inlet and the top boundary gives

$$\begin{aligned}
 f_1 &= f_3, \quad f_4 = f_2, \quad f_8 = f_6 \\
 f_5 &= f_7 = \frac{1}{2}[\rho - f_0 - 2(f_1 + f_2 + f_6)]
 \end{aligned} \tag{23}$$

Other corner nodes can be dealt with in a similar way.

2.4.3. Velocity boundary conditions. For velocity boundary conditions, both u_x and u_y are specified at the relevant flow boundaries. Then all the density functions at the boundary nodes are determined as in the previous pressure boundary case. A similar procedure can be followed, but the difference is that the fluid density at the boundary is solved first instead.

For the inlet (left) boundary, f_2 , f_3 , f_4 , f_6 and f_7 are known after streaming, while ρ and f_1 , f_5 , f_8 have to be solved. Following the procedure for the pressure boundary conditions, the solution is obtained as follows:

$$\begin{aligned}
 \rho_{\text{in}} &= \frac{1}{1 - u_x} [f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)] \\
 f_1 &= f_3 + \frac{2}{3}\rho_{\text{in}}u_x \\
 f_5 &= f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{\text{in}}u_x + \frac{1}{2}\rho_{\text{in}}u_y \\
 f_8 &= f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{\text{in}}u_x - \frac{1}{2}\rho_{\text{in}}u_y
 \end{aligned} \tag{24}$$

For the outlet (right) boundary,

$$\begin{aligned} \rho_{\text{out}} &= \frac{1}{1 + u_x} [f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)] \\ f_3 &= f_1 - \frac{2}{3}\rho_{\text{out}}u_x \\ f_6 &= f_8 - \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_{\text{out}}u_x + \frac{1}{2}\rho_{\text{out}}u_y \\ f_7 &= f_5 + \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_{\text{out}}u_x - \frac{1}{2}\rho_{\text{out}}u_y \end{aligned} \tag{25}$$

The treatment of corner nodes at the velocity flow boundaries is similar to that of the pressure boundaries as described in the previous subsection.

3. INCORPORATING A TURBULENCE MODEL INTO LB FORMULATION

Whether a fluid flow is laminar or not is often controlled by the Reynolds number defined by

$$Re = UL/\nu \tag{26}$$

where U is the typical macroscopic velocity of a flow and L is a characteristic length of the fluid domain. At high Reynolds numbers, a fluid enters turbulent regimes, becoming unsteady and non-linear. Turbulent flows are characterized by the occurrence of eddies with multiple scales in space, time and energy. The study of turbulent flows is one of the most active research areas in computational fluid dynamics (CFD) and a wide range of turbulence models have been put forward over the last one hundred years.

LES is a well-known turbulence methodology in the engineering field. It aims at directly solving large spatial-scale turbulent eddies that carry the majority of the energy, while modelling the smaller-scale eddies using a subgrid model. The separation of these scales is achieved through the filtering of the Navier–Stokes equations, from which solutions to the resolved scales are obtained. Unresolved scales can be modelled by, for instance, the widely used one-parameter Smagorinsky subgrid model [6] that assumes that the Reynolds stress tensor is dependent only on the local strain rate.

Though the LB method is well established for a variety of fluid flows, turbulence modelling within the LB framework remains a challenge and only very limited work has been reported. A simple route to the incorporation of turbulence modelling is to directly apply the concept of LES to the LB formulation. Following this approach, the filtered form of the LB equation is expressed as [10]

$$\tilde{f}_i(\mathbf{x} + \mathbf{e}_i\Delta t, t + \Delta t) = \tilde{f}_i(\mathbf{x}, t) - \frac{1}{\tau_*}[\tilde{f}_i(\mathbf{x}, t) - \tilde{f}_i^{\text{eq}}(\mathbf{x}, t)] \quad (i = 0, \dots, 8) \tag{27}$$

where \tilde{f}_i and \tilde{f}_i^{eq} represent, respectively, the distribution function and the equilibrium distribution function at the resolved scale. The effect of the unresolved scale motion is modelled through an effective collision relaxation time scale τ_t . Thus, in Equation (27) the total relaxation time should be

$$\tau_* = \tau + \tau_t$$

where τ and τ_t are, respectively, the relaxation times corresponding to the true fluid (molecular) viscosity ν and the turbulence viscosity ν_t defined by a subgrid turbulence model. Accordingly,

v_* is given by

$$v_* = v + v_t = \frac{1}{3}(\tau_* - \frac{1}{2})C^2\Delta t = \frac{1}{3}(\tau + \tau_t - \frac{1}{2})C^2\Delta t$$

$$v_t = \frac{1}{3}\tau_t C^2\Delta t$$

By employing the Smagorinsky model, the turbulence viscosity v_t is explicitly calculated from the filtered strain rate tensor $\tilde{S}_{ij} = (\partial_j \tilde{u}_i + \partial_i \tilde{u}_j)/2$ and a filter length scale (which is equal to the lattice spacing h) as

$$v_t = (Sc h)^2 \hat{S} \quad (28)$$

where Sc is the Smagorinsky constant and \hat{S} the characteristic value of the filtered strain rate tensor \tilde{S} :

$$\hat{S} = \sqrt{\sum_{i,j} \tilde{S}_{ij} \tilde{S}_{ij}}$$

An attractive feature of the model is that \tilde{S} can be obtained directly from the second-order moments, \tilde{Q} , of the non-equilibrium distribution function:

$$\tilde{S} = \frac{\tilde{Q}}{2\rho Sc \tau_*} \quad (29)$$

in which \tilde{Q} can be simply computed by the filtered density functions at the lattice nodes:

$$\tilde{Q}_{ij} = \sum_{k=1}^8 e_{ki} e_{kj} (\tilde{f}_k - \tilde{f}_k^{\text{eq}}) \quad (30)$$

where e_{ki} is the k th component of the lattice velocity \mathbf{e}_i . Consequently,

$$\hat{S} = \frac{\hat{Q}}{2\rho Sc \tau_*} \quad (31)$$

where \hat{Q} is the filtered mean momentum flux, computed from \tilde{Q} :

$$\hat{Q} = \sqrt{2 \sum_{i,j} \tilde{Q}_{ij} \tilde{Q}_{ij}} \quad (32)$$

The above approach is extremely convenient in terms of numerical implementations as it leaves the LB equation unchanged except for the use of a new turbulent-related viscosity τ_* . By using this turbulence LB equation with the Smagorinsky subgrid model, good results have been reported for simulations of a well-documented benchmark test case at $Re = 40\,000$ [22, 23]. Yu *et al.* [10] indicate that this extended LB method can accurately capture important features of the decaying homogeneous isotropic turbulence and is potentially a reliable computational tool for turbulence simulations.

4. INTERACTION BETWEEN MOVING PARTICLES AND FLUID FLOW

In the particle transport problems concerned, the solid particles are mainly driven by the hydrodynamic forces exerted by the fluid, and the body forces if considered. In the context of LB

methodology, fluid particles impact the boundary of the solid particle, and they exchange momentum to alter their motion. It is important to correctly model the interactions between fluid and solid particles so as to capture the essential physical behaviour of the problem under consideration. This requires a physically correct ‘no-slip’ boundary condition to impose at the interface between the fluid and the particle, i.e. the fluid adjacent to the particle surface should have the velocity identical to the particle surface.

For stationary particles the bounce-back rule (16) can be imposed to accomplish the ‘no-slip’ condition as mentioned earlier, and the interaction (impact) force can also be readily computed. It is, however, more difficult to impose the moving solid/fluid ‘no-slip’ condition, particularly to obtain impact forces.

4.1. Lattice representation of solid particles

The first step to model fluid and solid particle interaction is to represent the particle by lattice nodes. Figure 3 illustrates lattice discretization of a circular particle, where nodes interior and exterior to the particle are, respectively, the solid and fluid nodes as introduced earlier. These nodes are further classified into three categories: (1) fluid boundary node—a fluid node connected at least with one solid node; (2) solid boundary node—a solid node connected at least with one fluid node; and (3) interior solid node—a solid node not connected to any fluid node. A link between a fluid boundary node and a solid boundary node is called a boundary link. The surface of a solid particle is assumed to be located in the middle of the boundary links. Clearly, the stepwise lattice representation of the surface of a circular particle is neither accurate nor smooth unless a sufficiently small lattice spacing is used. More seriously, when the particle is in motion, its

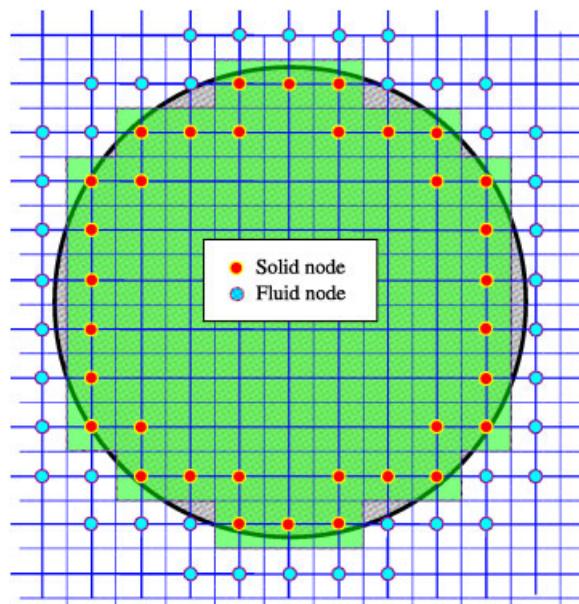


Figure 3. Lattice representation of a circular solid particle.

boundary nodes will continually change, but in an ‘on–off’ fashion, which has serious implications in the computed interaction forces, as will be discussed next. Also note that an effective algorithm needs to be in place to update the boundary nodes at each time step. Nevertheless, this discrete representation [24] provides a universal approach for any shaped particle; thus arbitrarily shaped particles can be readily handled within the current LB framework.

4.2. Hydrodynamic interaction force between fluid and solid particles

There are several approaches available for modelling the interaction between a moving particle and fluid, two of which are briefly discussed below.

4.2.1. Modified bounce-back rule. Ladd [5] proposed a modification to the original bounce-back rule so that the movement of a solid particle can be accommodated. Referring to Figure 4, for a given boundary link (i), the modified ‘no-slip’ rule is given by

$$f_{-i}(\mathbf{x}, t + 1) = f_i(\mathbf{x}, t_+) - \alpha_i \mathbf{e}_i \cdot \mathbf{v}_b \quad (\alpha_i = 6w_i \rho / C_s^2) \quad (33)$$

where $f_i(\mathbf{x}, t_+)$ represents the post-collision distribution at the (fluid or solid) boundary node \mathbf{x} , and \mathbf{v}_b is the velocity in the middle of the boundary link i computed as

$$\mathbf{v}_b = \mathbf{v}_c + \boldsymbol{\omega} \times \mathbf{r}_c \quad (\mathbf{r}_c = \mathbf{x} + \mathbf{e}_i \Delta t / 2 - \mathbf{x}_c)$$

in which \mathbf{v}_c and $\boldsymbol{\omega}$ are, respectively, the translational and angular velocities at the mass centre of the solid particle; \mathbf{x}_c and $\mathbf{x} + \mathbf{e}_i \Delta t / 2$ are, respectively, the coordinates of the centre and the (nominal) boundary point. Note that the modified rule (33) is applied in each streaming operation

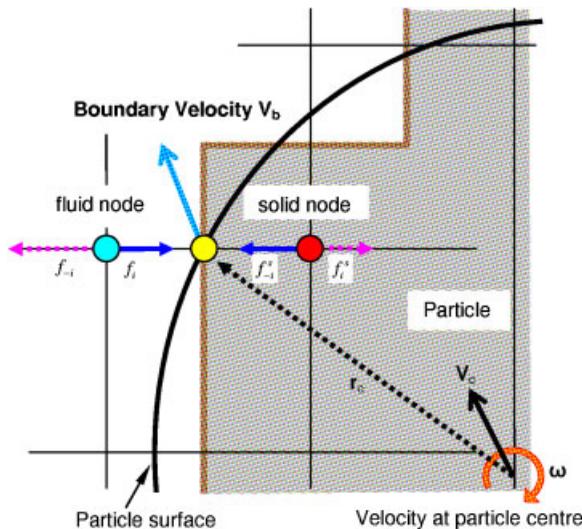


Figure 4. Modified bounce-back rule for a moving particle.

to both fluid and solid boundary nodes associated with the link. The new rule also implies that the fluid fills the entire solid particle or, in other words, the particle is modelled as a ‘shell’ filled with fluid. As a result, both solid and fluid nodes on either side of the boundary surface are treated in an identical fashion.

From the collision rule (33), the impact force on the solid particle from the link can be computed as

$$\mathbf{F}_i = 2[f_i(\mathbf{x}, t_+) - \alpha_i \mathbf{e}_i \cdot \mathbf{v}_b] / \Delta t \quad (34)$$

The corresponding torque \mathbf{T}_i , produced by the force with respect to the particle centre, is given by

$$\mathbf{T}_i = \mathbf{r}_c \times \mathbf{F}_i \quad (35)$$

Then the total hydrodynamic forces and torque exerted on the particle are obtained by summing up the forces and torques of all the related boundary links:

$$\mathbf{F} = \sum_i \mathbf{F}_i, \quad \mathbf{T} = \sum_i \mathbf{T}_i \quad (36)$$

There are two issues relevant to the modified ‘no-slip’ method: (1) The rule permits the transfer of fluid mass across the particle boundary and therefore facilitates the implementation, but at the same time it may have effects on particle inertia and solution accuracy [20]. The justification of the scheme is provided in [25]. (2) It has been observed that the computed interface force may suffer from severe oscillation when the particle moves across the grid with large velocity. This is mainly due to the time-changing ‘on-off’ nature of the boundary nodes in the discrete representation of solid particles. Some improvements in this respect have been suggested in [25]. It should be mentioned that Aidun *et al.* [20] propose an alternative approach that does not require the transfer of fluid into the solid particle and therefore no fluid is contained inside the solid boundary. Consequently, the exchange of momentum takes place only between the solid boundary and the adjacent fluid. The complexity of this method lies in the need to assign an appropriate initial density function at a fluid node that has just changed from a solid node at the last time step. The proposed method, however, can be adopted to tackle the problem with moving domain boundaries.

4.2.2. Immersed moving boundary method. The immersed moving boundary (IMB) method is a general numerical technique for the modelling of interaction between fluid and (deformable) structures. Attempts to incorporate the IMB concept into the LB formulation have been made, for instance, in [26, 27]. The scheme proposed by Noble and Torczynski [26] is adopted in this work. The basic ideas behind the approach include the establishment of a more accurate and smooth lattice representation of solid particles to reduce the fluctuation of the computed hydrodynamic forces, as well as the modification of the fluid density distribution at the nodes covered by a solid particle to enforce the ‘no-slip’ condition. It introduces a control volume/cell for each lattice node that is an $h \times h$ square around the node, as illustrated by the grey area in Figure 5(b). A local fluid/solid ratio γ is also defined, which is the area fraction of the nodal cell covered by a particle.

Then, the LB equation for those lattice nodes (fully or partially) covered by a solid particle is given by

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau} (1 - \beta) [f_i(\mathbf{x}, t) - f_i^{\text{eq}}] + \beta f_i^m \quad (37)$$

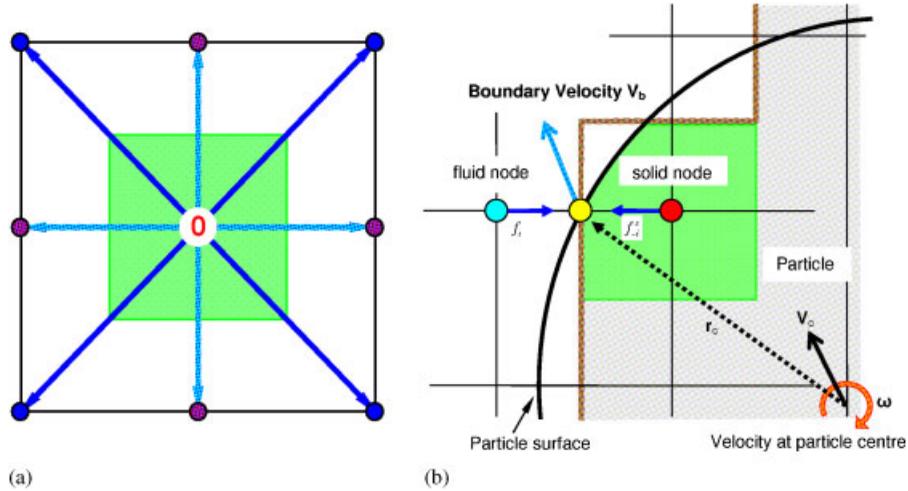


Figure 5. Immersed boundary scheme of Noble and Torczynski [26]: (a) control area of a node and (b) nodal solid area fraction.

where β is a weighting function depending on the local fluid/solid ratio γ ; f_i^m is an additional term that accounts for the bounce back of the non-equilibrium part of the distribution function. Three different forms of β and two possible formulations for f_i^m have been investigated in [26]. Their difference to the final result is generally not significant, which is confirmed by our numerical tests as well. Thus, the following forms of β and f_i^m are adopted in this work:

$$\beta = \frac{\gamma(\tau - 0.5)}{(1 - \gamma) + (\tau - 0.5)} \tag{38}$$

$$f_i^m = f_{-i}(\mathbf{x}, t) - f_i(\mathbf{x}, t) + f_i^{\text{eq}}(\rho, \mathbf{v}_b) - f_{-i}^{\text{eq}}(\rho, \mathbf{v})$$

The total hydrodynamic forces and torque exerted on a particle over n particle-covered nodes can be computed as [11, 26]

$$\mathbf{F}_f = Ch \left[\sum_n \left(\beta_n \sum_i f_i^m \mathbf{e}_i \right) \right] \tag{39}$$

$$\mathbf{T}_f = Ch \left[\sum_n (\mathbf{x}_n - \mathbf{x}_c) \times \left(\beta_n \sum_i f_i^m \mathbf{e}_i \right) \right] \tag{40}$$

where \mathbf{x}_n are the coordinates of the lattice node n . This particular version of the IMB scheme for the LB equations is slightly more complex in terms of implementation and computation than the modified bounce-back rule of (33). Nevertheless, the computed hydrodynamic forces are sufficiently smooth.

5. INTERACTIONS BETWEEN MOVING PARTICLES

When particles are not in direct contact among themselves but driven by fluid flow and body force (gravity), their motion can be determined by Newton's equations

$$\begin{aligned} m\mathbf{a} &= \mathbf{F}_f + m\mathbf{g} \\ J\ddot{\theta} &= \mathbf{T}_f \end{aligned} \quad (41)$$

where m and J are, respectively, mass and moment of inertia of the particle, $\ddot{\theta}$ is angular acceleration, \mathbf{g} is gravitational acceleration if considered, and \mathbf{F}_f and \mathbf{T}_f are, respectively, hydrodynamic forces and torque. The equations can be solved numerically by an explicit numerical integration, such as central difference scheme. When direct contact occurs between the particles and/or between particles and walls, their interactions should be dealt with appropriately. The natural route is to use the DEM to account for particle interactions, which leads essentially to a coupled LB–DE approach for fluid–particle problems.

5.1. Discrete element modelling of particle interaction

The DEM, since its origins in the 1970s [28], has emerged as an effective numerical solution procedure, particularly for problems comprising discrete objects. The DEM is based on the concept that individual material elements are considered to be separate and are (possibly) connected only along their boundaries by appropriate physically based interaction laws. The response of DEs depends on the interaction forces, and the dynamic equations governing the evolution of the system are normally solved by the explicit central difference time integration algorithm that is consistent with the LB formulation.

Conventionally, simple geometrical entities such as circular disks, ellipses and spheres are used to represent discrete objects. More complex shaped elements such as superquadrics and polygons/polyhedra [29–31] have also been introduced to more realistically represent objects encountered in practice.

In the context of the explicit integration scheme employed to solve the dynamic equations of the system, the major computational steps involved in the DEM at each time step consist of (global) spatial search, (local) interaction resolution, interaction force computation and solutions of element positions and velocities.

The global search aims to determine for each DE a list of neighbouring elements that may potentially interact with it. Significant progress has been made over the last years in the development of highly effective search algorithms [32]. A tree-based algorithm, ADST [33], is employed in the present work. In the second step, each potential contactor–target pair is locally resolved on the basis of their kinematic relationship, while any pair for which interaction does not occur is excluded from further processing. In the third step, the interaction forces between each contact pair are determined according to a constitutive relationship or interaction law. Finally, the velocities and displacements of the DEs are updated by the time integration scheme employed.

For the current problem, each particle is represented by a DE. A linear or Hertz contact model is employed for normal contact between a contact pair, while by friction between the particles is ignored. The hydrodynamic forces exerted by the fluid are also considered and the static buoyancy force is taken into account by reducing the gravitational acceleration to $(1 - \rho/\rho_s)\mathbf{g}$, where ρ_s is the density of a particle. However, as pointed out in [11], the DE contact model does not consider

the lubrication pressure developed between two approaching particles, which may be important for some applications. This issue deserves further investigation.

By taking into account all the forces acting on a solid particle, the dynamic equations can be symbolically expressed as (ignoring the equation for rotational motion for simplicity)

$$m\mathbf{a} + c\mathbf{v} = \mathbf{F}_c + \mathbf{F}_f + m\mathbf{g} \quad (42)$$

where \mathbf{F}_c denotes the total contact forces from other particles and/or the walls, c is a damping coefficient, and the term $c\mathbf{v}$ represents a viscous force that accounts for the effect of all possible dissipation forces existing in the system.

An important issue in the time integration of (42) is the determination of the critical time step due to the explicit nature of the central difference algorithm. For the linear normal contact model

$$F_c = k_n \delta \quad (43)$$

where k_n is the normal stiffness and δ the penetration, the critical time step is given by

$$\Delta t_{cr} = 2 \left(\sqrt{1 + \xi^2} - \xi \right) / \omega \quad (44)$$

where $\omega = \sqrt{k_n/m}$ is the local contact natural frequency and $\xi = c/2m\omega$ is the critical damping ratio. Note that this critical time-step may be too *optimistic* for impact systems [34] and that the fluid effect is not taken into account, which may further reduce the value. Consequently, the actual time-step used for the integration of Equation (42) is taken as

$$\Delta t_D = \lambda \Delta t_{cr} \quad (45)$$

where the time-step factor λ is chosen to be around 0.1 to ensure both stability and reasonable accuracy of the solution.

5.2. Coupling with the LB equation: subcycling and lattice form

When combining the above DE modelling of the particle interaction with the LB formulation, a number of computational issues arise, two of which, subcycling time integration and lattice form of the DE equation (42), are briefly discussed below.

5.2.1. Subcycling time integration. There are two time steps used in the coupled LB–DE procedure, Δt for the fluid flow and Δt_D for the particles. Since Δt_D may be in general smaller than Δt , it has to be reduced to a new value Δt_s so that the ratio between Δt and Δt_s is an integer n_s :

$$\Delta t_s = \frac{\Delta t}{n_s} \quad (n_s = \lceil \Delta t / \Delta t_D \rceil + 1) \quad (46)$$

where $\lceil \cdot \rceil$ denotes an integer round-off operator. This basically gives rise to a so-called subcycling time integration for the DE part; in one step of the fluid computation, n_s sub-steps of integration are performed for the DE equation (42) using the time step Δt_s , whilst the hydrodynamic forces \mathbf{F}_f are kept unchanged during the subcycling. Our numerical experiments show that this subcycling scheme generally works well for small n_s . Its behaviour for a large value of n_s is, however, unclear and further numerical tests are required. It is worth highlighting that the numerical stability of the coupled LB–DE method with subcycling is a very subtle issue and related work on the

stability analysis of general multiple time-step integrations for dynamic systems [35] may reveal the complex nature of the issue.

5.2.2. *The DE equation in the lattice system.* Since the LB equation is implemented in the lattice coordinate system in this work, the DE equation should be implemented in the same way. It can be derived that in the lattice coordinate system Equation (42) has the form

$$\bar{m}\bar{\mathbf{a}} + \bar{c}_d\bar{\mathbf{v}} = \bar{\mathbf{F}}_c + \bar{\mathbf{F}}_f + \bar{m}\bar{\mathbf{g}} \tag{47}$$

where

$$\begin{aligned} \bar{m} &= m/\rho_s h^2, & \bar{\mathbf{v}} &= \mathbf{v}/C \\ \bar{\mathbf{a}} &= \mathbf{a}\Delta t/C, & \bar{\mathbf{g}} &= \mathbf{g}\Delta t/C \\ \bar{c}_d &= Chc_d, & \bar{\mathbf{F}}_t &= \mathbf{F}_t/(\rho_0 C^2 h) \end{aligned}$$

For the linear normal contact model

$$\bar{\mathbf{F}}_c = \bar{k}_n \bar{\delta}, \quad \bar{k}_n = k_n/C^2$$

Then the natural frequency $\bar{\omega}$, the critical time step $\bar{\Delta}t_{cr}$ and the time step $\bar{\Delta}t_s$ are, respectively, taken as

$$\bar{\omega} = \omega\Delta t, \quad \bar{\Delta}t_{cr} = \Delta t_{cr}/\Delta t, \quad \bar{\Delta}t_s = 1/n_s$$

6. NUMERICAL ILLUSTRATIONS

The performance of the coupled LB–DE approach is demonstrated with the simulation of particle transport against gravity from the bottom of a fluid domain along a pipe under the suction action resulting from the negative pressure difference applied.

The problem is simplified to a fluid–particle interaction simulation. The details are illustrated in Figure 6. The two inclined lines represent the pipe boundaries. The fluid domain is divided into an 800×800 square lattice with spacing $h = 2.5$ mm. Though the fluid domain should be rectangular in the LB method, a polygonal fluid domain is taken as the actual computational domain to reduce the computational costs, since both left-top and right-bottom sub-domains can be excluded from the simulation. To accommodate this irregularity, the actual domain profile is identified first, and the LB equation (2) is applied only to the nodes within the profile. This is a generic approach that can be extended to any problem with an irregular exterior domain boundary. The material properties of the fluid are chosen as density $\rho = 1000$ kg/m³ and kinematic viscosity $\nu = 5 \times 10^{-5}$ m²/s.

A constant pressure boundary condition with $\rho_{in} = \rho$ is imposed to the two (inlet) boundaries as shown in the figure. A smaller pressure with $\rho_{out} = 0.97\rho$ is applied to the outlet of the pipe. The remaining boundaries are assumed stationary walls and thus the ‘no-slip’ boundary condition is imposed.

A total of 70 circular particles with different sizes uniformly distributed from 30 to 80 mm are randomly positioned at the bottom of the domain as shown, using the packing algorithm developed in [36]. Full gravity ($g = 9.81$ m/s²) is considered. The immersed boundary method of Noble and Torczynski [26] is employed to compute the fluid–particle interaction forces. Different versions of the method are compared and no significant difference is observed. The linear disk/disk and

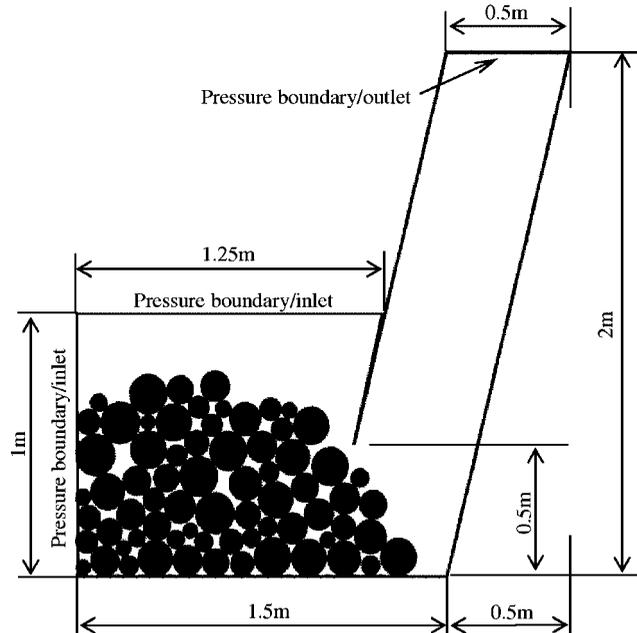


Figure 6. Example: problem description.

disk/segment contact models are used to model the contact between the particles and between particles and walls. The following parameters are chosen: particle density $\rho_s = 5000 \text{ kg/m}^3$, normal contact stiffness $k_n = 5 \times 10^8 \text{ N/m}$, contact damping ratio $\zeta = 0.5$ and time-step factor $\lambda = 0.1$, which gives a time step of $\Delta t_D = 3.37 \times 10^{-5}$ for the DE simulation of the particles.

In the first attempt, no turbulence model is employed in the simulation. In order to match the small viscosity of the fluid, the relaxation time τ should be, at least, around 0.505, but the corresponding LB formulation is unstable and no result can be obtained. Consequently, a turbulence model is incorporated in the simulation.

The LES-based one-parameter Smagorinsky turbulence model, as described in Section 3, is adopted with the Smagorinsky constant $Sc = 0.1$. A complete simulation is achieved with $\tau = 0.501$. This gives a time step $\Delta t = 4.17 \times 10^{-5} \text{ s}$ and thus the corresponding lattice speed $C = 60 \text{ m/s}$. The subcycle number n_s is computed as $n_s = 2$. The simulated maximum fluid velocity is approximately $v_{\max} = 5.6 \text{ m/s}$ at the pipe outlet (with the characteristic length $L = 0.5 \text{ m}$). The maximum Mach number and Reynolds number are therefore estimated as

$$Ma = \frac{v_{\max}}{C} = 0.0933, \quad Re = \frac{v_{\max} * L}{\nu} = 56\,000$$

The Mach number indicates that the results obtained are reasonably accurate. The flow field in terms of the total velocity contour and the evolution of the particles at four time instances are depicted in Figures 7(a)–(d), from which the complex fluid flow patterns due to fluid–particle interactions, particle/particle and particle/wall collisions are clearly observed. The general pattern and behaviour of the entire system appears to be realistic. Nevertheless, the results need further

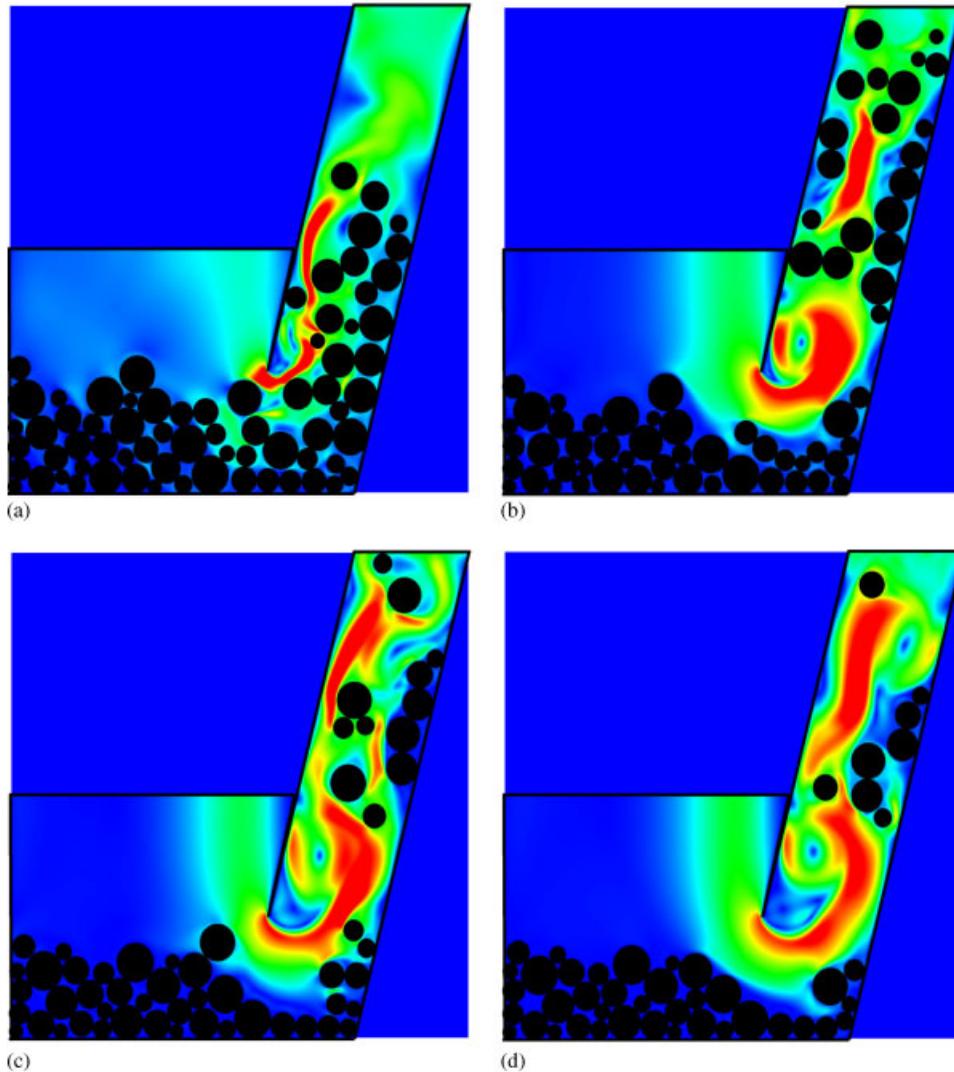


Figure 7. Example: total velocity contours of fluid flow at four time instances.

validations against experiment data, which has been conducted in a full 3D case and will be reported later.

7. CONCLUSIONS

This paper presents numerical procedures with respect to the coupled LB–DE methodology for the modelling of particle transport in incompressible fluid flows. Numerical tests confirm that the coupled approach, together with the incorporation of an appropriate turbulence model, is indeed

a promising solution strategy for the solution of particle–fluid interaction problems dominated by the presence of a large number of densely packed particles and the occurrence of turbulence. The method appears to be robust (up to a certain Reynolds number) and the implementation is simple. Model parameters can be easily selected. Further tests indicate that the number of particles and their size distribution can all be arbitrarily specified without causing any numerical problem. Different shapes of particles can also be readily accommodated into the LB–DE method, as reported in [37], with the use of the recently developed polygon/polygon contact model [29, 38].

The coupled methodology is described for 2D problems, but extension to 3D cases has already proven to be straightforward. Although the core LB operation (2) is very effective, the total computational cost can be substantial since a sufficiently fine lattice is often necessary and millions of time steps also need to be performed. It is therefore a dominant issue even for a moderate-sized 3D problem. This naturally leads to the utilization of parallel computing that is further promoted by the natural parallelism inherent in the LB formulation [39]. Note that the presence of moving particles may cause minor dynamic processor loading imbalance problems, which may be tackled by the dynamic domain decomposition parallel strategy developed in [40].

APPENDIX A: CONVERSIONS BETWEEN PHYSICAL AND LATTICE UNITS

Variable	Physical	Lattice	Relationship
Density	ρ	$\bar{\rho}$	$\rho = \rho_0 \bar{\rho}$
Density function	f_i	\bar{f}_i	—
Relaxation time	τ	$\bar{\tau}$	—
Spacing	h	$\bar{h} = 1$	—
Time step	Δt	$\bar{\Delta} t = 1$	—
Lattice speed	$C = \frac{h}{\Delta t}$	$\bar{C} = 1$	—
Viscosity	$\nu = \frac{1}{3}(\tau - \frac{1}{2})\frac{h^2}{\Delta t}$	$\bar{\nu} = \frac{1}{3}(\bar{\tau} - \frac{1}{2})$	$\nu = Ch\bar{\nu}$
Coor./Disp.	\mathbf{x}	$\bar{\mathbf{x}}$	$\mathbf{x} = h\bar{\mathbf{x}}$
Velocity	$\mathbf{v} = \dot{\mathbf{x}}$	$\bar{\mathbf{v}} = \frac{d\bar{\mathbf{x}}}{d\bar{t}}$	$\mathbf{v} = C\bar{\mathbf{v}}$
Acceleration	$\mathbf{a} = \frac{d^2\mathbf{x}}{dt^2}$	$\bar{\mathbf{a}} = \frac{d^2\bar{\mathbf{x}}}{d\bar{t}^2}$	$\mathbf{a} = \frac{C}{\Delta t}\bar{\mathbf{a}}, \mathbf{g} = \frac{C}{\Delta t}\bar{\mathbf{g}}$
Hydrodynamic force	$\mathbf{F}_t = Ch(\sum \beta_n \sum f_i^m \mathbf{e}_i)$	$\bar{\mathbf{F}}_t = \sum \bar{\beta}_n \sum \bar{f}_i^m \bar{\mathbf{e}}_i$	$\mathbf{F}_t = \rho_0 C^2 h \bar{\mathbf{F}}_t$
Mass	m	\bar{m}	$m = \rho_0 h^2 \bar{m} = h$
Stiffness	k_n	\bar{k}_n	$k_n = C^2 \bar{k}_n$
Critical time step	Δt_{cr}	$\bar{\Delta} t_{cr}$	$\Delta t_{cr} = \Delta t \bar{\Delta} t_{cr}$

REFERENCES

1. Kuipers J, van Duin K, van Beckum F, van Swaaij W. A numerical model of gas-fluidized beds. *Chemical Engineering Science* 1992; **47**:1913–1924.
2. Tsuji Y, Kawaguchi T, Tanaka T. Discrete particle simulation of two-dimensional fluidized bed. *Powder Technology* 1993; **77**(1):79–87.
3. Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering* 2001; **8**(2):83–130.

4. Chen S, Doolen G. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics* 1998; **30**:329–364.
5. Ladd A. Numerical simulations of fluid particulate suspensions via a discretized Boltzmann equation (Parts I & II). *Journal of Fluid Mechanics* 1994; **271**:285–339.
6. Smagorinsky J. General circulation experiments with the primitive equations: I. The basic equations. *Mon. Weather Review* 1963; **91**:99–164.
7. Chen H, Kandasamy S, Orszag S, Shock R, Succi S, Yakhot V. Extended Boltzmann kinetic equation for turbulent flows. *Science* 2003; **301**:633–636.
8. Hou S, Sterling J, Chen S, Doolen GD. A lattice Boltzmann subgrid model for high Reynolds number flows. *arXiv preprint comp-gas/9401004*, 1994.
9. Latt J, Chopard B, Succi S, Toschi F. Numerical analysis of the averaged flow field in a turbulent lattice Boltzmann simulation. *Physica A: Statistical Mechanics and its Applications* 2006; **362**(1):6–10.
10. Yu H, Girimaji S, Lu L. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *Journal of Computational Physics* 2005; **209**:599–616.
11. Cook BK, Noble DR, Williams JR. A direct simulation method for particle–fluid systems. *Engineering Computations* 2004; **21**(2–4):151–168.
12. Han K, Feng YT, Owen DRJ. Coupled lattice Boltzmann and discrete element modelling of particle–fluid problems. *Computers and Structures* 2007; **85**(11–14):1080–1088.
13. Qian Y, d’Humières D, Lallemand P. Lattice BGK models for Navier–Stokes equation. *Europhysics Letters* 1992; **17**:479–484.
14. Frisch U, d’Humières D, Hasslacher B, Lallemand P, Pomeau Y, Rivet J-P. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems* 1987; **1**:649–707.
15. Lallemand P, Luo L-S. Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability. *Physical Review E* 2000; **61**:6546–6562.
16. Sterling JD, Chen S. Stability Analysis of Lattice Boltzmann Method. *Journal of Computational Physics* 1996; **123**:196–206.
17. Bouzidi M, d’Humières D, Lallemand P, Luo L-S. Lattice Boltzmann equation on a two-dimensional rectangular grid. *Journal of Computational Physics* 2001; **172**:704–717.
18. Filippova O, Hänel D. Acceleration of lattice-BGK schemes with grid refinement. *Journal of Computational Physics* 2000; **165**:407–427.
19. Zou Q, He X. On pressure and velocity flow boundary conditions for the lattice Boltzmann BGK model. *Los Alamos Preprint* 1995; **9**(8).
20. Aidun CK, Lu Y, Ding E-G. Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation. *Journal of Fluid Mechanics* 1998; **373**:287–311.
21. Ginzburg I, d’Humières D. Multireflection boundary conditions for lattice Boltzmann models. *Physical Review E* 2003; **68**:066614.
22. Rodi W, Ferziger JH, Breuer M. Status of large eddy simulation: results of a workshop. *Journal of Fluids Engineering (ASME)* 1997; **119**(2):248–262.
23. Krafczyk M, Toelke J. Large-eddy simulations with a multiple-relaxation-time LBE model. *International Journal of Modern Physics B* 2003; **17**(1–2):33–39.
24. Bouzidi M, Firdaouss M, Lallemand P. Momentum transfer of a Boltzmann-Lattice fluid with boundaries. *Physics of Fluids* 2001; **13**(11):3452–3459.
25. Ladd A, Verberg R. Lattice-Boltzmann simulations of particle–fluid suspensions. *Journal of Statistical Physics* 2001; **104**(5/6):1191–1251.
26. Noble D, Torczynski J. A lattice Boltzmann method for partially saturated cells. *International Journal of Modern Physics C* 1998; **9**:1189–1201.
27. Feng Z-G, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid particles interaction problems. *Journal of Computational Physics* 2004; **195**:602–628.
28. Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. *Geotechnique* 1979; **29**(1):47–65.
29. Feng YT, Owen DRJ. A 2D polygon/polygon contact model: algorithmic aspects. *Engineering Computations* 2004; **21**:265–277.
30. Feng YT, Han K, Owen DRJ. An energy based polyhedron-to-polyhedron contact model. *Proceedings of 3rd M.I.T. Conference of Computational Fluid and Solid Mechanics*, MIT, U.S.A., 14–17 June 2005.
31. Owen DRJ, Feng YT, DeSouza Nato E, Wang F, Cottrell MG, Pires FA, Yu J. The modeling of multi-fracturing solids and particulate media. *International Journal for Numerical Methods in Engineering* 2004; **60**(1):317–340.

32. Williams JR, Perkins E, Cook B. A contact algorithm for partitioning N arbitrary sized objects. *Engineering Computations* 2004; **21**:215–234.
33. Feng YT, Owen DRJ. An augmented spatial digital tree algorithm for contact detection in computational mechanics. *International Journal for Numerical Methods in Engineering* 2002; **55**:556–574.
34. Feng YT. On the central difference algorithm in discrete element modeling of impact. *International Journal for Numerical Methods in Engineering* 2005; **64**(14):1959–1980.
35. Feng YT, Han K, Owen DRJ. Asynchronous/multiple time integrators for impact and discrete systems. *Proceedings of 8th International Conference on Computational Plasticity (COMPLAS VIII)*, Barcelona, Spain, 5–8 September 2005.
36. Feng YT, Han K, Owen DRJ. Filling domains with disks: an advancing front approach. *International Journal for Numerical Methods in Engineering* 2003; **56**:699–713.
37. Han K, Feng YT, Owen DRJ. Numerical simulations of irregular particle transport in turbulent flows using coupled LBM–DEM. *Computer Modeling in Engineering and Science* 2007; **18**(2):87–100.
38. Han K, Feng YT, Owen DRJ. Contact resolution for non-circular discrete objects. *International Journal for Numerical Methods in Engineering* 2006; **66**(3):485–501.
39. Harting J, Chin J, Venturoli M, Coveney PV. Large-scale lattice Boltzmann simulations of complex fluids: advances through the advent of computational Grids. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 2005; **363**(1833):1895–1915.
40. Owen DRJ, Feng YT. Parallelised finite/discrete element simulation of multi-fracture solids and discrete systems. *Engineering Computations* 2001; **18**(3/4):557–576.