Computers and Fluids 249 (2022) 105690

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

The immersed boundary method for confined flows: Numerical diffusion and simulation accuracy of a boundary retraction scheme

Alessia Abbati [a], Ya Zhang [b], William Dempster [a], Yonghao Zhang [c,d,*]

[a] Department of Mechanical and Aerospace Engineering, University of Strathclyde, G1 1XJ, Glasgow, UK
[b] China Automotive Innovation Corporation, Nanjing, Jiangsu, 211100, China
[c] School of Engineering, The University of Edinburgh, Edinburgh, EH9 3FD, UK
[d] Institute of Mechanics, Chinese Academy of Sciences, Beijing, 100190, China

ARTICLE INFO

Keywords:
Immersed boundary method
Lattice Boltzmann method
Diffuse-interface
Boundary retraction

ABSTRACT

Diffuse-interface immersed boundary methods (IBM) have been successfully applied to numerous complex fluid–structure interaction problems because of their simple and efficient implementation. Despite their ability to suppress numerical oscillations significantly compared to sharp-interface methods, the diffuse interface is likely to reduce simulation accuracy of the flow field around the solid boundary. The present work investigates the diffusive effects of IBM and the mitigation method for surface-confined particulate flows by comparing results to sharp-interface methods. It is found that increasingly-confined geometries accentuate interface diffusion effects and decrease simulation accuracy. To minimise the diffusive effects of IBM, a boundary retraction scheme is used and its effectiveness is examined, in particular for particles in close contact where the diffuse interfaces overlap. It is shown that this simple implementation is capable of alleviating interface diffusion errors, thus increasing accuracy while limiting computational costs. With an optimal boundary retraction scheme, IBM can successfully capture fluid–structure interactions at different degrees of confinement, comparable to sharp-interface methods.

1. Introduction

The *immersed boundary method* (IBM) is an efficient implementation of fluid–structure boundary conditions that simplifies the mesh generation process for complex arbitrary geometries and does not require expensive re-meshing strategies for moving and deformable objects. The basic idea of the IBM is to solve the flow field on a body non-conforming Eulerian grid and represent the fluid–structure interface with a set of Lagrangian marker points. The no-slip condition on the boundary is enforced by introducing an appropriate force term to the governing equations. Since it was first introduced by Peskin [1], the IBM has been extensively applied to biofluid dynamics, multiphase flows, porous media, particle suspensions, acoustics, and especially particulate flows, which are characterised by complex dynamic interactions between individual particles, the surrounding fluid and solid surfaces [2,3].

In the original formulation of the IBM [1], the additional force is evaluated with a *feedback* process based on the deformation of the boundary marker points, which are modelled as elastic springs. However, this method is incapable of modelling perfectly rigid boundaries and introduces a free parameter that affects the stability and restricts the computational time step [4,5]. In the *direct-forcing* formulation, the boundary force is computed at the Lagrangian marker points by imposing the known boundary velocity that matches the fluid velocity at that position [6,7]. However, this force evaluation strategy may not be able to accurately satisfy the no-slip boundary condition for the forcing point velocity [4,8,9]. In the approach suggested by Wu and Shu [10], the unknown forces are computed implicitly by solving a linear matrix equation to enforce the no-slip condition at the boundary points. Wang et al. [9] proposed a *multi-direct-forcing* method, which implements an iterative forcing process to solve the linear equation and achieves the no-slip condition at all boundary points without affecting the computational efficiency [5].

Generally, the Eulerian grid and Lagrangian marker points exchange information through velocity interpolation and force distribution. The operations between the different node systems are based on discrete delta functions, which diffuse the interface around the solid boundary [2], as interpolation stencils involving discrete delta functions formally introduce a first-order error term in the velocity field [2, 11–14]. Another consequence of the diffuse interface is the *effective thickness* of the fluid–solid boundary, which changes the apparent size

[*] Corresponding author at: School of Engineering, The University of Edinburgh, Edinburgh, EH9 3FD, UK.
E-mail address: yonghao.zhang@imech.ac.cn (Y. Zhang).

https://doi.org/10.1016/j.compfluid.2022.105690
Received 22 May 2022; Received in revised form 14 September 2022; Accepted 4 October 2022
Available online 7 October 2022

of solid objects and surface locations [15–19]. The diffusive effect becomes increasingly significant for confined flows and particle collisions, which are common features for particulate flows.

Although the original IBM was devised as a numerical method solving the Navier–Stokes equation, it was later applied to the *lattice Boltzmann method* (LBM) framework. Based on a mesoscopic fluid description, LBM has become a viable alternative to the Navier–Stokes solvers for complex flows owing to its simple formulation and efficient parallel computations. The *immersed boundary-lattice Boltzmann method* (IB-LBM) was first proposed by Feng and Michaelides [20] and was then extended to a multi-direct-forcing method by Suzuki and Inamuro [21].

Alternatively, in the LBM framework there are several methods that feature a sharp representation of the interface as a fluid–solid boundary treatment. The most common strategies are the simple and interpolated bounce-back methods [22,23], which suffer from spurious numerical oscillations for moving boundary problems. The *partially saturated method* presented by Noble and Torczynski [24] is a modified bounce-back approach that allows for a sharp reconstruction of the boundary by representing lattice nodes as pure fluid, pure solid, and mixed nodes depending on a weighting parameter. To improve the accuracy of this sharp-interface method, Chen et al. [25] combined it with a *ghost method* that introduces virtual nodes within the solid object to extrapolate fluid properties. The method presented in the work of Tsigginos et al. [26] achieves second-order accuracy by modelling fluid–particle interaction as a two-phase mixture. The solid objects are represented as virtual fluids and the computational nodes are identified as pure fluid, pure virtual fluid, and partially saturated based on their solid fraction. However, accurate calculation of the solid fraction is pivotal to correctly represent complex fluid–solid interfaces. Therefore, this method introduces a significant computational cost for moving boundary problems as the solid fraction needs to be updated at every time step.

To take advantage of IBM while alleviating the diffuse interface effects on the flow field, Breugem [11] suggested to correct the effective diameter of immersed objects by retracting the Lagrangian interface grid points some distance inside the solid surface. It is shown that this strategy partly cancels the interpolation errors and improves the simulation accuracy significantly. Additionally, the retraction distance is affected by the solid object geometry, and the optimal values for spheres, tubes and cubes are provided. Since discrete delta functions used in IBM have different interpolation ranges, the effective boundary thickness is also affected by the choice of kernel function. The work of Peng et al. [13] tested the boundary retraction approach for different discrete delta functions and concluded that the optimal distance value is larger for more diffusive functions. Furthermore, Peng and Wang [14] derived an analytical demonstration of the optimal retraction distance for a plane channel flow. However, for surface confined particulate flows in particular with inter-particle collisions, much work is required to understand the extent to which the diffuse interface affects simulation accuracy and to improve the boundary retraction approach accordingly.

The present work applies the relaxed multi-direct-forcing IBM proposed by Zhang et al. [27], which introduces an estimated optimal relaxation parameter to accelerate the convergence rate and improve the enforcement of the no-slip boundary condition. Moreover, the IBM is coupled with the *cascaded lattice Boltzmann method* (CLBM) of Geier et al. [28] as it improves numerical stability by performing collisions in the moment space using central moments [29]. The main focus of this study is to investigate the diffuse interface effects for particulate flows with no geometrical confinement, fixed and variable confinements, and particle interactions and collisions. The results are compared to experimental data in Ref. [30] and the results of sharp-interface methods in Refs. [26,31] to highlight the effects of different interface representations.

The rest of the paper is structured as follows. Section 2 presents a brief introduction to the cascaded lattice Boltzmann method and the IB-LBM algorithm. In Section 3, the numerical method is applied to four benchmark problems: flow past a cylinder, settling of a particle under gravity, flow around two impacting particles, and a drafting–kissing–tumbling process of two particles settling in a channel. The results are compared to other experimental and numerical studies, and the effect of the diffuse interface is systematically analysed. Concluding remarks are drawn in Section 4.

2. Numerical methods

2.1. Lattice Boltzmann method

Based on a discrete form of the Boltzmann model equation, the LBM provides a mesoscopic description of fluid transport that tracks fictitious particles residing on the nodes of a uniform lattice. The velocity domain is restricted to a discrete velocity set $\mathbf{c}_\mathrm{i}$ whose component vectors directly connect neighbouring lattice nodes with lattice spacing Δx. The velocity set used in this work is the D2Q9 model, which features a two-dimensional lattice of nine lattice velocities. The velocity vectors $\mathbf{c}_\mathrm{i} = \left\{c_\mathrm{ix},\, c_\mathrm{iy} \mid i = 0, \dots, 8\right\}$ are defined as follows

$$\begin{aligned} c_\mathrm{ix} &= [0, 1, 0, -1, 0, 1, -1, -1, 1]\ , \\ c_\mathrm{iy} &= [0, 0, 1, 0, -1, 1, 1, -1, -1]\ . \end{aligned} \tag{1}$$

The *discrete-velocity distribution function* $f_\mathrm{i}(\mathbf{x}, t)$ represents the number of particles with discrete velocity $\mathbf{c}_\mathrm{i}$ at position $\mathbf{x}$ and time t. The lattice Boltzmann equation describes the evolution of the discrete-velocity distribution function in the following way

$$f_\mathrm{i}(\mathbf{x} + \mathbf{c}_\mathrm{i}\,\Delta t, t + \Delta t) = f_\mathrm{i}(\mathbf{x}, t) + \Omega_i(\mathbf{x}, t)\Delta t + S_i(\mathbf{x}, t)\Delta t\,, \tag{2}$$

where Δt is the time step, Ω_i is the collision operator, and S_i is the force term.

The general LBM algorithm is divided into two consecutive parts, *collision* and *streaming*, which are respectively expressed as

$$f_\mathrm{i}^\star(\mathbf{x}, t) = f_\mathrm{i}(\mathbf{x}, t) + \Omega_i(\mathbf{x}, t)\Delta t + S_i(\mathbf{x}, t)\Delta t\,, \tag{3a}$$

$$f_\mathrm{i}(\mathbf{x} + \mathbf{c}_\mathrm{i}\,\Delta t, t + \Delta t) = f_\mathrm{i}^\star(\mathbf{x}, t)\,, \tag{3b}$$

where $f_\mathrm{i}^\star$ is the post-collision discrete-velocity distribution function. The collision process in Eq. (3a) rearranges the distribution functions $f_\mathrm{i}(\mathbf{x}, t)$ at each lattice node. In Eq. (3b), the distribution functions after collision $f_\mathrm{i}^\star(\mathbf{x}, t)$ are streamed along the lattice linking with velocity $\mathbf{c}_i$ to a neighbouring point.

The operator Ω_i models particle collisions by performing appropriate relaxation processes towards equilibrium. In the CLBM, collisions are performed in the moment space using the central moments of the distribution function, which are obtained in a frame of reference moving with the macroscopic fluid velocity [28]. The CLBM collision operator Ω_i can be modelled as $\Omega_i = \mathbf{K} \cdot \mathbf{k}$ where $\mathbf{k}$ is the moment space of the distribution function f_i and $\mathbf{K}$ is an orthogonal transformation matrix. The central moments are relaxed in ascending order with individual relaxation rates [32–34]. To achieve an isotropic viscous stress tensor, it is required that $\omega_4 = \omega_5 = \omega_\nu$, where ω_ν determines the kinematic shear viscosity [32]. All the other relaxation parameters are set to unity [32]. The shear viscosity ν is defined as

$$\nu = c_s^2 \left(\frac{1}{\omega_\nu} - \frac{1}{2} \right)\,, \tag{4}$$

where $c_s^2 = 1/\sqrt{3}$ is the lattice speed of sound. Additionally, the source term S_i incorporates the macroscopic body force $\mathbf{F}$, which is implemented with a split-forcing approach [27,35].

The macroscopic density $\rho(\mathbf{x}, t)$ and velocity $\mathbf{u}(\mathbf{x}, t)$ are respectively the zeroth and first moments of the discrete-velocity distribution functions. Their expressions are given by

$$\rho(\mathbf{x}, t) = \sum_i f_\mathrm{i}(\mathbf{x}, t)\,, \tag{5}$$

$$\rho\mathbf{u}(\mathbf{x}, t) = \sum_i \mathbf{c}_\mathrm{i}\, f_\mathrm{i}(\mathbf{x}, t) + \frac{1}{2}\mathbf{F}(\mathbf{x}, t)\Delta t\,. \tag{6}$$

2.2. Immersed boundary method

IB-LBM algorithms implement two different computational node systems. The lattice nodes $\mathbf{x}$ define the fixed regular Eulerian grid on which the fluid equations are solved. The fluid–solid interface is represented by a set of Lagrangian marker points $\mathbf{x}_\ell$ that are not bound to the underlying lattice and are allowed to move in space. The no-slip boundary condition is enforced by introducing an appropriate force term $\mathbf{F}_\mathrm{b}$ that mimics the effect of the boundary. The boundary force modifies the fluid velocity to ensure that it matches the known boundary velocity $\mathbf{u}_\ell$ at the Lagrangian marker positions $\mathbf{x}_\ell$, i.e.

$$\mathbf{u}(\mathbf{x}_\ell, t) = \mathbf{u}_\ell(\mathbf{x}_\ell, t) \, . \tag{7}$$

The fluid velocity at the Lagrangian marker positions is calculated by interpolating the fluid velocity at the neighbouring lattice points

$$\mathbf{u}(\mathbf{x}_\ell, t) = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x}, t)\, \Phi(\mathbf{x} - \mathbf{x}_\ell)\, \Delta x^2 \, , \tag{8}$$

where the interpolation kernel Φ is an appropriate discretised version of the Dirac delta distribution

$$\Phi(\mathbf{x}) = \frac{1}{\Delta x^2}\, \phi\left(\frac{x}{\Delta x}\right) \phi\left(\frac{y}{\Delta x}\right) . \tag{9}$$

Several kernel functions for Φ with different interpolation range have been implemented in the literature [2,13,27]. It has been demonstrated that kernel functions with larger support intervals are more effective at suppressing numerical fluctuations but result in significant diffusive effects on the fluid–solid interface [13]. In this work the most widely-used four-point regularised discrete delta function ϕ_4 [2] is considered because of its numerical stability [13,27], i.e.

$$\phi_4(r) = \begin{cases} \frac{1}{8}\left(3 - 2\,|r| + \sqrt{1 + 4\,|r| - 4r^2}\right) , & 0 \le |r| \le 1 \, , \\ \frac{1}{8}\left(5 - 2\,|r| - \sqrt{-7 + 12\,|r| - 4r^2}\right) , & 1 \le |r| \le 2 \, , \\ 0 \, , & 2 \le |r| \, , \end{cases} \tag{10}$$

where r is the parameter representing the position of the submerged boundary point and is scaled with respect to the lattice length.

Here, we use the IBM model presented in Zhang et al. [27] where an appropriate boundary force term $\mathbf{F}_\mathrm{b}$ is introduced to satisfy the no-slip boundary condition as expressed by Eq. (7). Based on the velocity expression given by Eq. (6), the corresponding velocity correction $\delta\mathbf{u}$ is calculated as follows [27],

$$\rho(\mathbf{x}, t)\,(\mathbf{u}(\mathbf{x}, t) + \delta\mathbf{u}(\mathbf{x}, t)) = \sum_i \mathbf{c}_i f_\mathrm{i}(\mathbf{x}, t) + \frac{1}{2}\left(\mathbf{F}(\mathbf{x}, t) + \mathbf{F}_\mathrm{b}(\mathbf{x}, t)\right) \Delta t \, . \tag{11}$$

Therefore, the immersed boundary force $\mathbf{F}_\mathrm{b}$ can be expressed as

$$\mathbf{F}_\mathrm{b}(\mathbf{x}, t) = \frac{2\rho\, \delta\mathbf{u}(\mathbf{x}, t)}{\Delta t} \, . \tag{12}$$

Similarly, the corresponding immersed boundary force on the Lagrangian points $\mathbf{F}_\ell$ can be obtained in the following way

$$\mathbf{F}_\ell(\mathbf{x}_\ell, t) = \frac{2\rho\, \delta\mathbf{u}_\ell(\mathbf{x}_\ell, t)}{\Delta t} \, , \tag{13}$$

where $\delta\mathbf{u}_\ell$ is the velocity correction on the Lagrangian points.

To evaluate the Eulerian body force density $\mathbf{F}_\mathrm{b}$, the force acting on the Lagrangian points $\mathbf{F}_\ell$ is distributed to the lattice nodes next to the boundary using

$$\mathbf{F}_\mathrm{b}(\mathbf{x}, t) = \sum_{\mathbf{x}_\ell} \mathbf{F}_\ell(\mathbf{x}_\ell, t)\, \Phi(\mathbf{x} - \mathbf{x}_\ell)\, \Delta s \, , \tag{14}$$

where Δs is the spacing between the Lagrangian markers. Considering the boundary force expressions given by Eqs. (12) and (13), the force distribution operation in Eq. (14) can be evaluated in the following way

$$\delta\mathbf{u}(\mathbf{x}, t) = \sum_{\mathbf{x}_\ell} \delta\mathbf{u}_\ell(\mathbf{x}_\ell, t)\, \Phi(\mathbf{x} - \mathbf{x}_\ell)\, \Delta s \, . \tag{15}$$

The interpolation and distribution procedures described respectively in Eqs. (8) and (15) are the source of the diffusive effects due to the implementation of discrete delta functions [13].

The hydrodynamic force $\mathbf{F}_\mathrm{h}$ and torque $\mathbf{T}_\mathrm{h}$ acting on a solid object are calculated by

$$\mathbf{F}_\mathrm{h}(\mathbf{x}, t) = -\sum_{\mathbf{x}_\ell} \mathbf{F}_\ell(\mathbf{x}_\ell, t)\, \Delta s + \mathbf{F}_\mathrm{i} \, , \tag{16}$$

$$\mathbf{T}_\mathrm{h}(\mathbf{x}, t) = -\sum_{\mathbf{x}_\ell} \left(\mathbf{x}_\ell - \mathbf{X}\right) \times \mathbf{F}_\ell(\mathbf{x}_\ell, t)\, \Delta s + \mathbf{T}_\mathrm{i} \, , \tag{17}$$

where $\mathbf{F}_\mathrm{i}$ and $\mathbf{T}_\mathrm{i}$ are respectively the force and torque resulting from the internal fluid mass [29], and $\mathbf{X}$ is the centre of mass of the solid object.

In contrast to the diffuse-interface IBM, the partially saturated method (PSM) provides an alternative sharp-interface representation [24]. PSM approximates complex boundaries by identifying lattice nodes as pure fluid, pure solid, and partially saturated according to their solid fraction $0 \le \varepsilon \le 1$. As a consequence, the interface thickness is limited to one lattice. In the PSM algorithm, the standard LBM equation given by Eq. (2) is modified in the following way

$$\begin{aligned} f_\mathrm{i}(\mathbf{x} + \mathbf{c}_\mathrm{i}\, \Delta t,\, t + \Delta t) = {} & f_\mathrm{i}(\mathbf{x}, t) + (1 - B)\, \Omega_i\, \Delta t + B\, \Omega_i^\mathrm{s} \Delta t \\ & + (1 - B)\, S_i(\mathbf{x}, t) \Delta t \, , \end{aligned} \tag{18}$$

where Ω_i^s is the collision operator for solid nodes and B is a numerical weighting parameter based on the solid fraction ε. A significant disadvantage of the PSM is the cumbersome process of calculating correct values for the solid fraction ε, which has to be updated at every time step for moving boundary problems.

Within the PSM framework, Tsigginos et al. [26] model solid objects and the surrounding fluid as a two-phase mixture by introducing a virtual fluid with infinite viscosity in place of solid bodies. A momentum exchange algorithm (MES-I) is used to account for the presence of the fluid–solid interface in order to enforce the no-slip condition on the virtual fluid phase velocity. Since the velocity of partially saturated nodes is also dependent on the fluid phase velocity, to improve MES-I, another momentum exchange algorithm (MES-II) is implemented by directly enforcing the solid object velocity on both virtual fluid phase and partially saturated nodes. This work also derives a second-order accurate scheme for calculation of the body force, which resolves inaccuracy caused by the weighting parameter B formulation in the original PSM [26]. Because of their accuracy, the PSM results will be used in this work to evaluate the performance of the diffuse-interface IBM.

3. Numerical simulations

In contrast to sharp-interface approaches, the accuracy of the IB-LBM scheme deteriorates quickly as particulate flow becomes more confined due to the diffusive boundary effect. To alleviate this, the simple but effective boundary retraction strategy is applied here. The performance is evaluated in four cases: flow past a cylinder, settling of a particle under gravity, flow around two impacting particles, and drafting–kissing–tumbling process of two particles settling in a channel. These test cases are arranged to show how the diffusive effect of IBM influences flows with very different confinements and to examine the effectiveness of the boundary retraction scheme. The simulation results will be compared with experimental data, and numerical results obtained with the sharp-interface approach PSM in Ref. [26].

3.1. Flow past a cylinder

The flow past a stationary circular cylinder in a two-dimensional channel is used to test the performance of the IB-LBM scheme for fixed boundary problems in both confined and unconfined geometries.

Table 1
Geometry and fluid properties for unconfined flow past a cylinder.

Geometry		Parameters	
Diameter D	1	Density ρ_f	1
Square domain width L	50	Viscosity μ_f	0.001

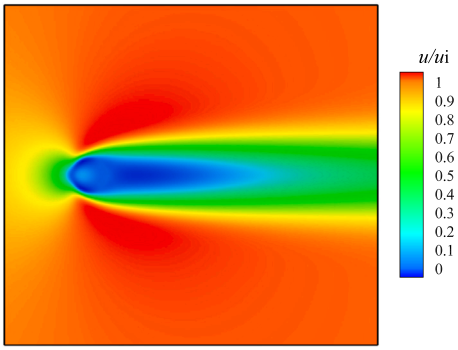


Fig. 1. Contours of the dimensionless horizontal velocity u/u_i for unconfined flow past a cylinder at $Re = 40$.

3.1.1. *Unconfined flow*

The stationary circular cylinder of diameter D is positioned at the centre of a square of size L. A fixed constant velocity u_i is prescribed at inlet, top and bottom boundaries, while a pressure boundary condition is imposed at the outlet. The results from extensive experiments of Tritton [30] and the sharp-interface approach PSM of Tsigginos [26] are employed for a quantitative comparison.

The geometry parameters and fluid properties listed in Table 1 are normalised the same as Tsigginos et al. [26] for consistent comparison. The Reynolds number is increased from 1 to 40 by varying the inlet velocity u_i according to the equation $Re = \rho_f u_i D/\mu_f$, where ρ_f and μ_f are respectively the density and the dynamic viscosity of the fluid. The drag coefficient C_d is calculated from

$$C_d = \frac{2F}{\rho_f u_i^2 D} \,. \tag{19}$$

A lattice size $\Delta x = D/30$ is used in the current case as it was demonstrated to be sufficiently accurate [26].

The contours of the normalised horizontal velocity u/u_i are displayed in Fig. 1. The results are shown for $Re = 40$, representing the corresponding regime characterised by a steady wake with separated flow. The calculated drag coefficients C_d for different Re values are plotted in Fig. 2 showing a good agreement with both the experimental data [30] and the numerical results [26]. Therefore, for the unconfined flow above, the diffuse interface does not lead to significant numerical error, and the boundary retraction scheme is not required.

3.1.2. *Confined flow and wall interaction effects*

The flow past a cylinder of diameter D in a channel of width W and length L is simulated to investigate the confinement effect on the flow behaviour in the creeping laminar flow regime, where the bounding walls of the channel alter the characteristic features of the flow field. The channel confinement ratio $\kappa = D/W$ is the characteristic parameter for measuring the extent of confinement. This classical case has been studied extensively and detailed description of the available analytical studies, computational simulations and experiments have been presented in several works, e.g. Refs. [36–39].

The computational domain geometry and the fluid field are displayed in Fig. 3. A parabolic velocity profile with a maximum speed of u_m is imposed at the inlet and outlet boundaries. No-slip boundary conditions are applied at the top and bottom channel walls with the halfway bounce-back scheme. The geometry and fluid properties used in the simulations are listed in Table 2.

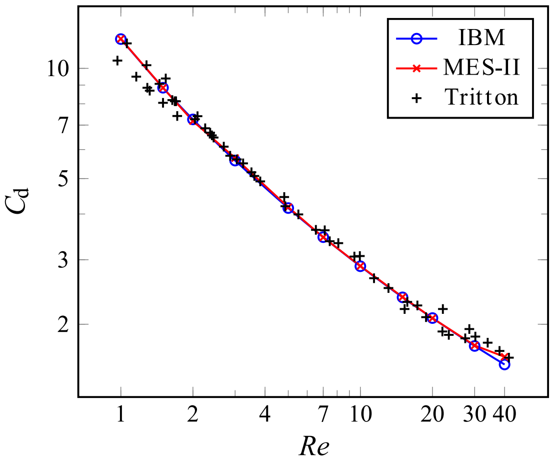


Fig. 2. Drag coefficient C_d for unconfined flow past a cylinder at different Reynolds numbers Re. Experimental results from Tritton [30] and MES-II numerical results in Ref. [26] are given for comparison.

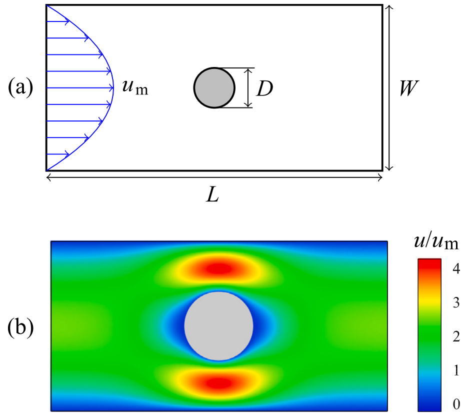


Fig. 3. Flow past a fixed cylinder: (a) schematic of the computational domain; (b) contours of the dimensionless horizontal velocity u/u_m.

For the current flow past a cylinder in the creeping laminar flow regime, the wall effect on the drag force can be estimated using a *wall correction factor* λ, which is defined as

$$\lambda(\kappa) = \frac{F_d(\kappa)}{u_m \mu_f} \,, \tag{20}$$

where F_d is the drag force, and μ_f is the fluid viscosity. As one of the most common expressions for the wall correction factor that can offer an accurate analytical solution for Stokes flow, the wall correction factor λ_F derived by Faxén [26] is expressed in the form [26,36]

$$\lambda_F(\kappa) = \frac{4\pi}{f(\kappa) + g(\kappa)} \,, \tag{21}$$

where $f(\kappa)$ and $g(\kappa)$ are defined as

$$\begin{aligned} f(\kappa) &= A_0 - \left(1 + 0.5\,\kappa^2 + A_4\,\kappa^4 + A_6\,\kappa^6 + A_8\,\kappa^8\right)\ln(\kappa) \,, \\ g(\kappa) &= B_2\,\kappa^2 + B_4\,\kappa^4 + B_6\,\kappa^6 + B_8\,\kappa^8 \,, \end{aligned} \tag{22}$$

and the corresponding numerical coefficients are

$$\begin{aligned} A_0 &= -0.9156892732 \,, & B_2 &= 1.26653975 \,, \\ A_4 &= 0.05464866 \,, & B_4 &= -0.9180433 \,, \\ A_6 &= -0.26462967 \,, & B_6 &= 1.8771010 \,, \\ A_8 &= 0.792986 \,, & B_8 &= -4.66549 \,. \end{aligned}$$

To evaluate the performance of the IB-LBM scheme, the calculated wall correction factor λ is compared to the Faxén wall correction factor

Table 2
Geometry and fluid properties for the flow past a cylinder simulations.

Geometry		Parameters	
Diameter D	1	Density ρ_f	1
Channel width W	2.5	Viscosity μ_f	0.005
Channel height H	40	Velocity u_m	10^{-6}
Confinement ratio κ	0.4	Reynolds number Re	2×10^{-4}

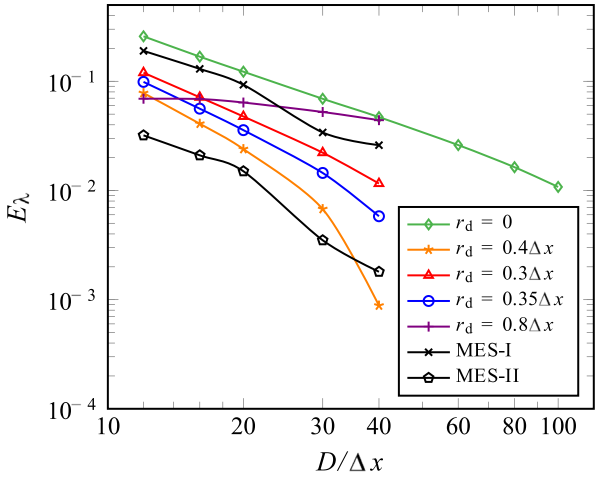


Fig. 4. Wall correction factor error E_λ for confined flow past a cylinder for the confinement ratio $\kappa = 0.4$. The numerical results (MES-I, MES-II) reported in Ref. [26] are given for comparison.

λ_F. The numerical error E_λ is given by the expression,

$$E_\lambda = \left| \frac{\lambda - \lambda_F}{\lambda_F} \right| . \tag{23}$$

Simulations are performed for different lattice sizes, i.e. $\Delta x = (D/12, D/16, D/20, D/30, D/40)$ to show the effect of the computational grid. The corresponding numerical error E_λ with $\kappa = 0.4$ is plotted in Fig. 4 and compared to the results in Ref. [26]. As Fig. 4 shows, the simulation results accuracy increases drastically for a larger lattice resolution. Importantly, the accuracy of the present simulation is comparable to the MES-I scheme, but it is less accurate than the second-order algorithm MES-II. It seems that the diffuse interface in the highly-confined geometry ($\kappa = 0.4$) has a more significant impact on the performance of the IB-LBM scheme compared to sharp-interface methods including MES-I and MES-II.

The approach of retracting the Lagrangian surface grid points is applied to the simulations with the retraction distance $r_d = (0, 0.3, 0.35, 0.4)\Delta x$, where the case with $r_d = 0$ does not include boundary retraction. As displayed in Fig. 4, the wall correction factor error E_λ decreases considerably as r_d grows. The error E_λ with $r_d = 0.35\Delta x$ decreases significantly from 4.7% to 0.58% for the lattice size $\Delta x = D/40$ and from 26% to 9.9% for $\Delta x = D/12$. It can also be noted that retracting the Lagrangian points by $r_d = 0.4\Delta x$ for lattice size $\Delta x = D/40$ makes the results even more accurate than the ones for $\Delta x = D/100$ without retraction. The above results suggest an appropriate $r_d = 0.4\Delta x$, which is close to the optimal value suggested by Breugem [11] and Peng et al. [14]. Additionally, a larger retraction distance $r_d = 0.8\Delta x$ is also considered in Fig. 4. It is clear that $r_d = 0.8\Delta x$ performs significantly worse. The effect of the retraction distance is further investigated by expanding the value range to $0 \le r_d \le 0.8$ with an interval of 0.1. The corresponding error results E_λ are presented in Fig. 7. It is demonstrated that the error progressively reduces when the boundary retraction scheme is used, reaching a minimum for $r_d = 0.4\Delta x$.

Since a larger degree of confinement accentuates interface diffusion in the IB-LBM simulations, it is interesting to compare the results of different confinement ratios $\kappa = (0.1, 0.2, 0.4)$, where the corresponding channel widths W are (10, 5, 2.5). By comparing the cases in Figs. 4–6, it can be seen that the wall correction factor error E_λ becomes much

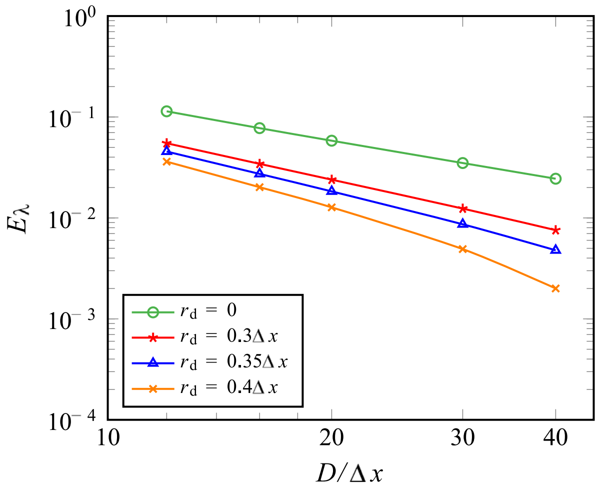


Fig. 5. Wall correction factor error E_λ for confined flow past a cylinder for the confinement ratio $\kappa = 0.2$.

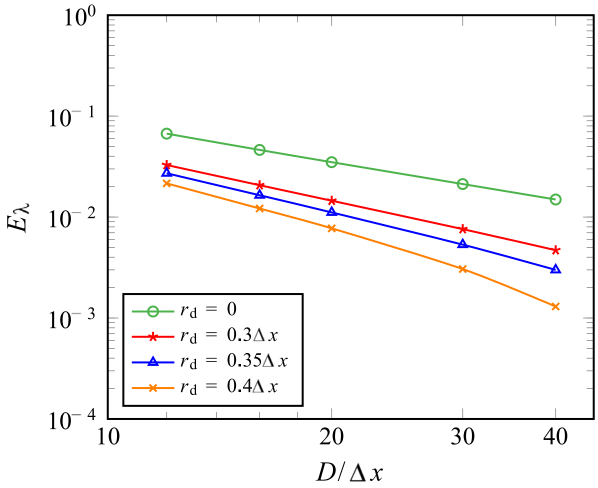


Fig. 6. Wall correction factor error E_λ for confined flow past a cylinder for the confinement ratio $\kappa = 0.1$.

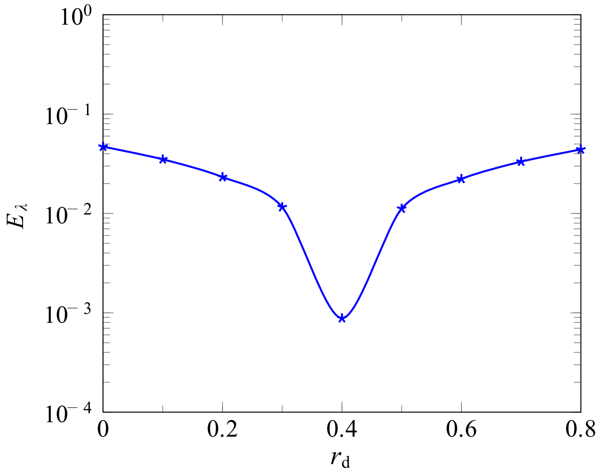


Fig. 7. Confined flow past a cylinder: variation of the wall correction factor error E_λ with the retraction distance r_d for confinement ratio $\kappa = 0.4$.

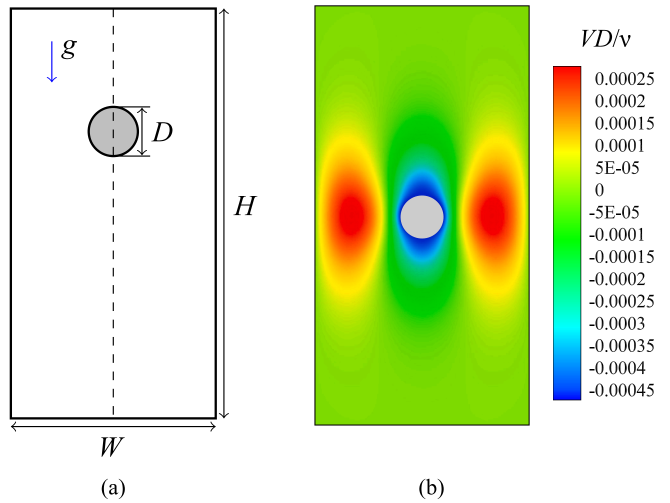


Fig. 8. Sedimentation of a particle under gravity: (a) schematics of the computational domain and (b) contours of the vertical velocity V.

Table 3
Particle and fluid properties for sedimentation of a particle under gravity.

Geometry		Parameters	
Diameter D	1	Density ρ_f	1
Channel width W	5	Viscosity μ_f	0.005
Channel height H	80	Particle density ρ_s	2.6
Confinement ratio κ	0.2	Reynolds number Re	5×10^{-4}

smaller as the confinement ratio κ decreases. Specifically for lattice size $\Delta x = D/40$, when the confinement ratio κ is reduced from 0.4 to 0.1, the error E_λ for cases without retraction decreases from 4.7% to 1.5%, while E_λ for cases with $r_d = 0.35\Delta x$ reduces from 0.58% to 0.3%.

As demonstrated in Figs. 4–6, the interface diffusion effect can be compensated successfully by applying different retraction distances, $r_d = (0.3, 0.35, 0.4)\Delta x$. To be specific, as the retraction distance r_d grows from zero to $0.4\Delta x$, in Fig. 5 with $\kappa = 0.2$, the error E_λ reduces from 2.4% to 0.2% for lattice sizes $\Delta x = D/40$ respectively, while in Fig. 6 with $\kappa = 0.1$, E_λ decreases from 1.5% to 0.13% for $\Delta x = D/40$. However, the retraction distance $r_d = 0.8\Delta x$ makes the accuracy of the IBM much worse for all the tested confinement ratios κ. Therefore, the boundary retraction strategy is competitive in reducing computational cost while maintaining simulation accuracy, as long as an appropriate retraction distance is used.

3.2. Sedimentation of a single particle under gravity

The case of a cylindrical particle settling under gravity in a two-dimensional channel is simulated to investigate confined flow with moving boundaries in the creeping laminar flow regime. In addition to approximate analytical formulations [26], this problem has been investigated by various numerical and experimental studies, and extensive benchmark results are available [36,40–45].

As represented in Fig. 8, the problem geometry is schematised by a vertical channel of width W and height H. The computational domain is bounded by solid walls and a no-slip boundary condition is implemented using a halfway bounce-back scheme. The initial geometry and fluid flow conditions used in the simulations are listed in Table 3. A cylindrical particle of diameter D initially at rest is released along the centreline of the channel. As the particle starts moving downwards under the action of gravity, the drag force exerted by the surrounding fluid increases. Once the drag force balances the gravitational force, the particle settles with a constant velocity known as the *terminal velocity* V_t.

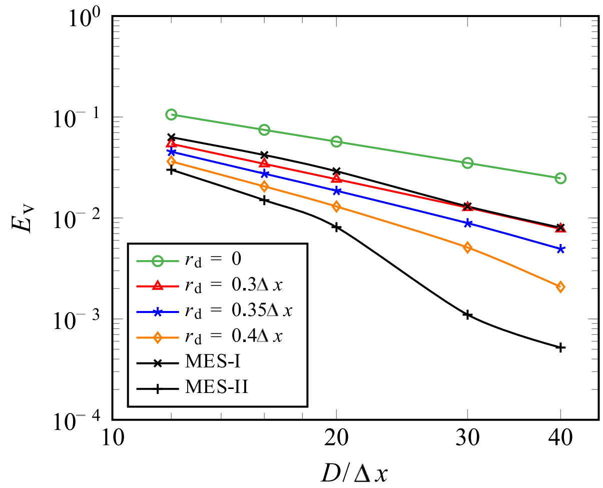


Fig. 9. Velocity error E_v for sedimentation of a particle under gravity for the confinement ratio $\kappa = 0.2$. The numerical results in Ref. [26] are given for comparison.

Because of the interaction with the walls, the settling velocity is lower than the corresponding one in the unconfined flow case [20,42,45]. A wall correction factor λ_F is introduced to account for the effect of the bounded geometry. By balancing the gravitational force $F_g = \left(\pi g D^2 \left(\rho_s - \rho_f\right)\right)/4$ and the drag force acting on the particle $F_d = \lambda_F \mu_f V_t$, the wall-corrected terminal velocity of a settling particle can be estimated as [26,45]

$$V_t = \frac{\pi g D^2 \left(\rho_s - \rho_f\right)}{4\mu_f \lambda_F(\kappa)}, \tag{24}$$

where ρ_s is the particle density, g is the gravitational acceleration, and λ_F is the Faxén wall correction factor. For this case, the Faxén wall correction factor is calculated as [26,46]

$$\lambda_F = \frac{4\pi}{A_0 - \ln(\kappa) + A_2\kappa^2 + A_4\kappa^4 + A_6\kappa^6 + A_8\kappa^8}, \tag{25}$$

where κ is the channel confinement ratio and the numerical coefficients are $A_0 = -0.9156892732$, $A_2 = 1.7243844$, $A_4 = -1.730194$, $A_6 = 2.405644$, and $A_8 = 4.59131$. The gravitational acceleration value g is calculated using Eq. (24) to match the specified value of the Reynolds number $Re = \rho_f V_t D/\mu_f$, where V is the numerical value of the terminal velocity. As the terminal velocity V_t can be accurately calculated by Eq. (24), it is used to evaluate the numerical velocity error E_v in the diffusive IB-LBM scheme as follows,

$$E_v = \left|\frac{V - V_t}{V_t}\right|. \tag{26}$$

Simulations are run for different lattice sizes, i.e. $\Delta x = (D/12, D/16, D/20, D/30, D/40)$. The calculated E_v for different lattice sizes is illustrated in Fig. 9 and compared to the numerical results of the sharp-interface method reported in Ref. [26]. The boundary retraction scheme with distance $r_d = (0.3, 0.35, 0.4)\Delta x$ is applied to alleviate the diffusive effects. Fig. 9 shows that the velocity error E_v decreases significantly when the boundary retraction approach is used. It can be estimated that the velocity error E_v for retraction distance $r_d = 0.4\Delta x$ decreases from 11% to 3.6% for the lowest resolution case $\Delta x = D/12$ and from 2.5% to 0.21% for lattice size $\Delta x = D/40$. The IB-LBM with retracted interface shows a better performance compared to the MES-I scheme, but the accuracy is still lower than the second-order MES-II model.

To investigate the effect of confinement, the particle sedimentation case is simulated in a wider computational domain with the confinement ratio $\kappa = 0.1$. The values of velocity error E_v for lattice sizes $\Delta x = (D/12, D/16, D/20, D/30, D/40)$ and the retraction distance $r_d = (0, 0, 3, 0.35, 0.4)\Delta x =$ are shown in Fig. 10. As observed in Fig. 10, the terminal velocity error E_v for the case of $r_d = 0.4\Delta x$ is reduced from

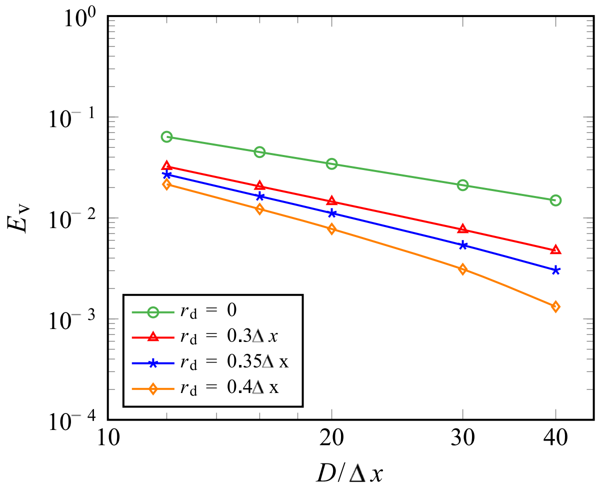


Fig. 10. Velocity error E_v for sedimentation of a particle under gravity for the confinement ratio $\kappa = 0.1$.

Table 4
Particle and fluid properties for flow around the two impacting particles.

Geometry		Parameters	
Diameter D	1	Density ρ_f	1
Channel width W	50	Viscosity μ_f	0.0015
Channel height H	50	Velocity U	0.15
Initial gap G	2	Reynolds number Re	50

1.5% to 0.13% for $D = 40\Delta x$ and from 6.4% to 2.2% for $D = 12\Delta x$. Moreover, it can be noted that due to the less significant confinement effect, the error E_v for $r_d = 0.4\Delta x$ in comparison to its corresponding case in Fig. 9 decreases from 3.6% to 2.2% and from 0.21% to 0.13% for $D = 12\Delta x$ and $D = 40\Delta x$, respectively. In this case, the retraction distance $r_d = 0.4\Delta x$ still shows the best accuracy.

3.3. Flow around two impacting particles

The test case of flow around two impacting cylindrical particles in a two-dimensional channel has been chosen to investigate the effect of diffuse interface for moving objects with a variable confinement and considering the interaction between particles where two diffuse interfaces could overlap. This problem has been studied by Bampalas and Graham [47] using a finite element method with a body-fitted mesh.

The computational domain used for the simulations is a channel of width W and length L, as illustrated in Fig. 11a. Two identical particles of diameter D are placed at the centre of the channel with an initial horizontal separation G. The fluid is initially stationary and the particles are moving towards each other along the centreline of the channel with the same velocity $U/2$. The computational domain is bounded by four stationary walls, where gradient boundary condition is imposed. Geometry and fluid properties used in this case are the same as in Ref. [26], which are listed in Table 4. Figs. 11b and c show the velocity contours at different instants, where the normalised time t^* is defined by the following expression [26,47],

$$t^* = \frac{U\,t - G}{D}\,, \tag{27}$$

where U is the relative velocity of the particles.

Simulations are carried out for different lattice sizes, i.e. $\Delta x = (D/12,\ D/30,\ D/60,\ D/80)$. Fig. 12 presents the evolution in time of the drag coefficient C_d for the different computational grids. As shown in Fig. 12, the accuracy of the IB-LBM model improves significantly by increasing the lattice resolution from $D = 12\Delta x$ to $D = 30\Delta x$, which is favoured. However, in spite of the great computational cost when increasing the lattice grid refinement to $D = 60\Delta x$ and $D = 80\Delta x$, the error on the drag coefficient C_d can hardly be much smaller.

The numerical error on the drag coefficient C_d appears to be related to the interface diffusion effect associated with the IBM model. Considering Fig. 11, it is evident that as the particles approach each other, the diffuse interface has an increasingly significant impact on the results. Therefore, the boundary retraction approach is used to minimise the interface diffusion effect, and improve simulation accuracy. The impact of this strategy can be visualised in Fig. 13 for three different grid resolutions, i.e. $D = (12, 30, 60)\Delta x$. It is evident that the results obtained using the boundary retraction scheme show a much close match to the ones presented in Ref. [26,47].

The difference between the drag coefficient C_d values for $r_d = 0$ and $r_d = 0.8\Delta x$ can be quantified by calculating the relative difference E_{Cd}. Fig. 13 shows the evolution of E_{Cd} as a function of the separation distance G/D between the particles for $D/\Delta x = [12,\ 30,\ 60]$. Initially, the drag coefficient C_d values for $r_d = 0$ and $r_d = 0.8\Delta x$ are similar and E_{Cd} is negligible. As the separation distance G/D between the particles is progressively reduced, the results for $r_d = 0$ and $r_d = 0.8\Delta x$ diverge and E_{Cd} becomes significant. In particular, when the separation distance G becomes smaller than $4\Delta x$, the diffuse interfaces start to overlap, and the numerical error for the drag coefficient C_d grows exponentially. As shown in Fig. 13, although the retraction distance $r_d = 0.4\Delta x$ works well for single-particle problems, a much larger retraction distance is needed, i.e. $r_d = 0.8\Delta x$ when the interaction between different diffuse interfaces becomes much stronger.

Table 5
Particle and fluid properties for the sedimentation of two particles simulations.

Geometry		Parameters	
Diameter D	0.2 cm	Density ρ_f	1 g/cm^3
Channel width W	2 cm	Viscosity μ_f	0.08 cm^2/s
Channel height H	6 cm	Particle density ρ_s	1.1 g/cm^3
Initial gap G	0.3 cm	Gravitational acceleration g	981 cm/s^2
Initial separation S	0.006 cm		

3.4. Drafting–kissing–tumbling process of two particles settling in a channel

The sedimentation of two circular cylindrical particles in a two-dimensional channel is considered to investigate moving boundary problems with a variable confinement. Since this test case involves a close interaction including collision between the falling particles, the diffuse interfaces could have a significant influence on their movement.

The fundamental mechanism associated with pairwise coupling of particles settling in a channel was first described by Fortes et al. [48] who performed experiments on spherical particles in fluidised beds. Significant work has been conducted to investigate this problem using various lattice Boltzmann models [49–51] and different IBM schemes [20, 52,53].

As shown in Fig. 14, the computational domain is geometrically represented by a vertical channel of width W and height H. Two identical particles of diameter D are placed along the centreline of the channel with an initial vertical separation G. The trailing particle is initially displaced off-centre with a horizontal separation distance from the centreline S. Non-slip boundary conditions are prescribed at the channel walls by applying a halfway bounce-back scheme. The computational domain geometry and fluid flow conditions used in the simulations are listed in Table 5, and these non-dimensional values are the same as Ref. [31].

Two particles settling in close proximity in a channel under gravity undergo a typical sequence of movements, known as the *drafting–kissing–tumbling* (DKT) process, which can be visualised in Fig. 14. After the particle are released, the trailing particle is caught in the wake of the leading particle and it experiences a reduction in its drag force. In the *drafting* part of the process, the trailing particle velocity

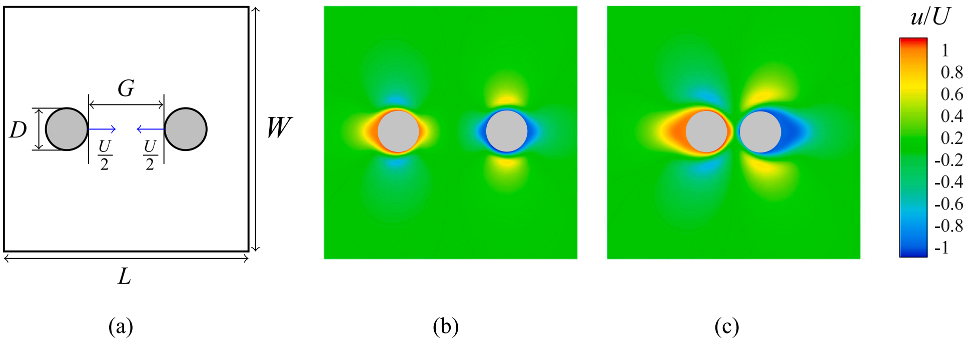


Fig. 11. Flow around two impacting cylinders: (a) schematics of the computational domain; (b) contours of the horizontal velocity U at $t^* = -1$; and (c) $t^* = -0.1$.

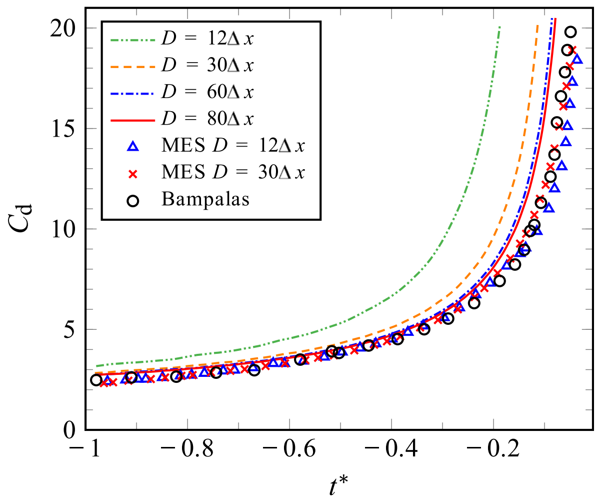


Fig. 12. Drag coefficient C_d evolution in time for the flow around two impacting cylinders. The numerical results in Ref. [26] (MES) and Ref. [47] (Bampalas) are given for comparison.

increases and it starts to fall faster than the leading one. The trailing particle eventually catches up with the leading one in the *kissing* stage, when the pair of particles are in contact. Because this configuration is unstable, the trailing particle rotates around the leading one and overtakes it in a *tumbling* movement.

Since the DKT process involves a contact stage, the IB-LBM implements the discrete element method to handle collision between particles. In the soft-sphere approach used, the interaction between particles is approximated by constructing the contact force as a function of the overlap and the relative velocity at the contact point. A linear spring–dashpot model is applied to both particle–particle and particle–wall interactions when the distance is less than the critical value ε_c. The contact force $\mathbf{F}_c$ can be decomposed into normal F_{cn} and tangential F_{ct} terms that are respectively calculated as follows,

$$F_{\rm cn} = k_{\rm n}\,\delta_{\rm n} + c_{\rm n}\,v_{\rm n}\,, \tag{28a}$$

$$F_{\rm ct} = \begin{cases} k_{\rm t}\,\Delta\delta_{\rm t} + c_{\rm t}\,v_{\rm t} & F_{\rm ct} < \mu\,F_{\rm ct}\,, \\ \mu\,F_{\rm ct} & F_{\rm ct} \geq \mu\,F_{\rm ct}\,, \end{cases} \tag{28b}$$

where k_n, k_t are the normal and tangential spring coefficients, c_n, c_t are the normal and tangential damping coefficients, v_n, v_t are the normal and tangential relative velocity at contact point, δ_n is the normal overlap, $\Delta\delta_t$ is the increment of the relative tangential displacement, and μ is the friction coefficient. In the present work, the critical value ε_c is set to $2\Delta x$ to accommodate the range of the four-point discrete delta function [29].

Simulation results are illustrated in Fig. 15, where the evolution of the velocity of the particles is plotted and compared to the sharp-interface method presented by Ardekani and Rangel [31]. Since the separation distance between the two particles for most of the DKT process approaches zero, the effect of the interface diffusion associated with the IBM becomes increasingly significant and the numerical accuracy is affected. In Fig. 15, the diffusive effect is alleviated by introducing the retraction distance scheme with $r_d = 0.4\Delta x$ and $r_d = 0.8\Delta x$, which can achieve accuracy equivalent to an increased spatial resolution.

From the above case for two impacting particles, we show that when the two particles are not in close contact, the usual r_d of $0.4\Delta x$ can be chosen. However, when the two particles move close to each other, the value of r_d should be switched to $0.8\Delta x$ as the diffuse interfaces start to overlap (G becomes $4\Delta x$ or less). To make computation simple, we test the retraction distance r_d of $0.4\Delta x$ and $0.8\Delta x$ in the current DKT process, and both show satisfactory results. However, the collision model used in the simulations has a significant influence on the interface retraction scheme. The present DEM scheme allows particle overlap by modelling their interaction using a linear spring–dashpot model, which is considerably different from the strategy used in Ref. [31]. Further work is required to improve the interface retraction technique.

4. Conclusion

Because of its versatility, diffuse-interface IBM is a viable method to simulate complex fluid–solid interaction problems. An important drawback of this approach is the diffuse interface, which changes both the apparent size of solid object and the exact surface location. In this work, the coupled IB-LBM has been applied to four benchmark problems to investigate the effects of interface diffusion. It has been found that the diffuse interface has negligible impact on unbounded geometries. However, in confined computational domains, surface interaction enhances interface diffusion effects and reduces the numerical accuracy of diffuse-interface IBM compared to sharp-interface methods. Interactions between particles is another challenging issue for traditional diffusive IBM formulations. When particles are in close contact, diffuse interfaces start to overlap which leads to more pronounced effects.

A boundary retraction approach has been tested to compensate for these sources of error. This strategy has been proved effective in achieving a better representation of the solid boundary and improving the results accuracy with limited computational costs. It has been demonstrated that compared to a traditional implementation, multi-direct-forcing IBM with boundary retraction achieves a better performance for confined geometries and particle interaction problems. The easy-to-use retraction distance is found numerically to be $r_d = 0.4\Delta x$, which is to be doubled when two diffuse interfaces start to overlap.

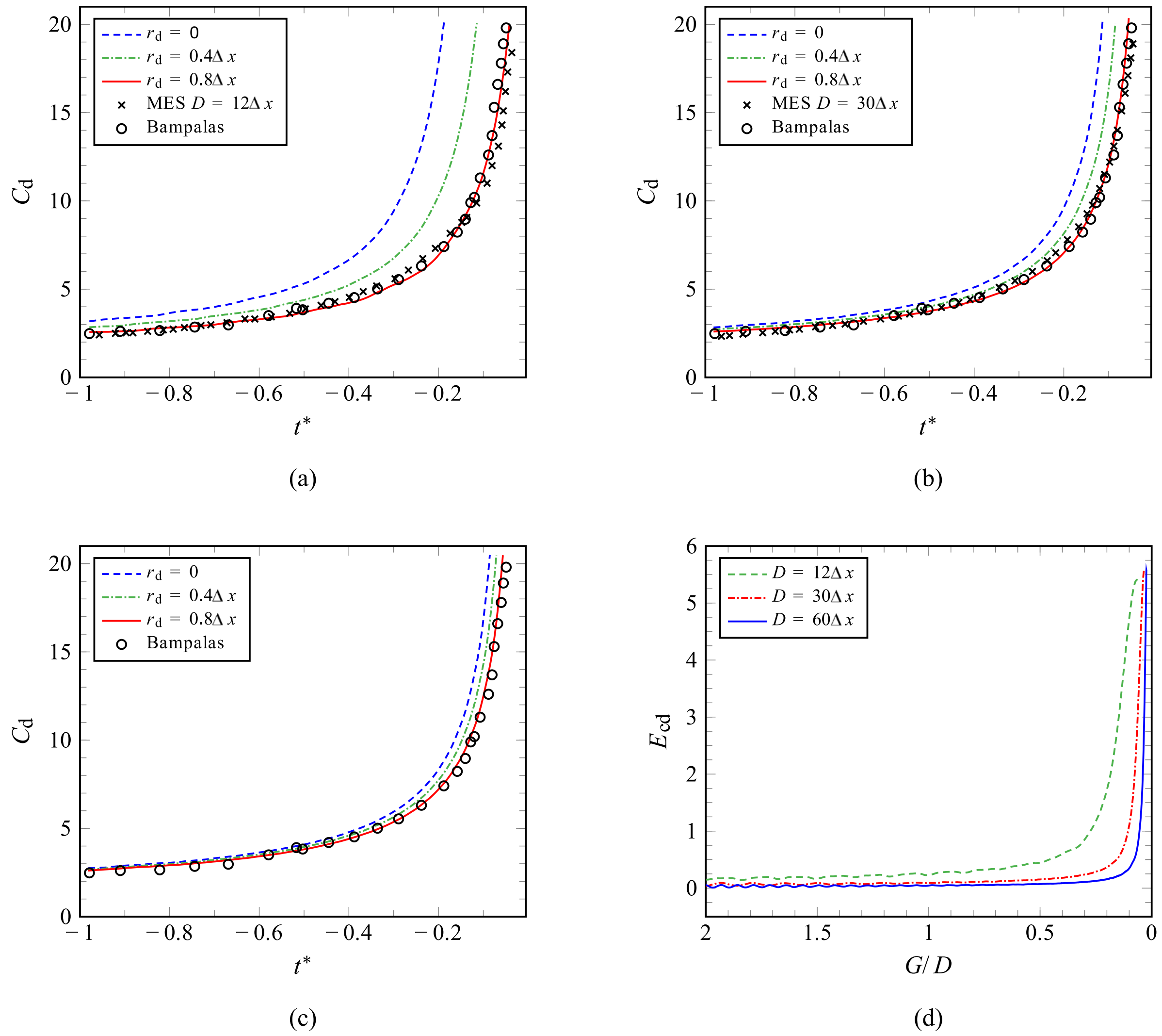


Fig. 13. Flow around two impacting cylinders: effect of the diffuse-interface thickness on the drag coefficient C_d evolution in time for lattice sizes: (a) $\Delta x = D/12$, (b) $\Delta x = D/30$, and (c) $\Delta x = D/30$. The numerical results in Ref. [26] (MES) and Ref. [47] (Bampalas) are given for comparison. (d) Drag coefficient difference E_{Cd} as a function of the separation distance between the particles G/D.

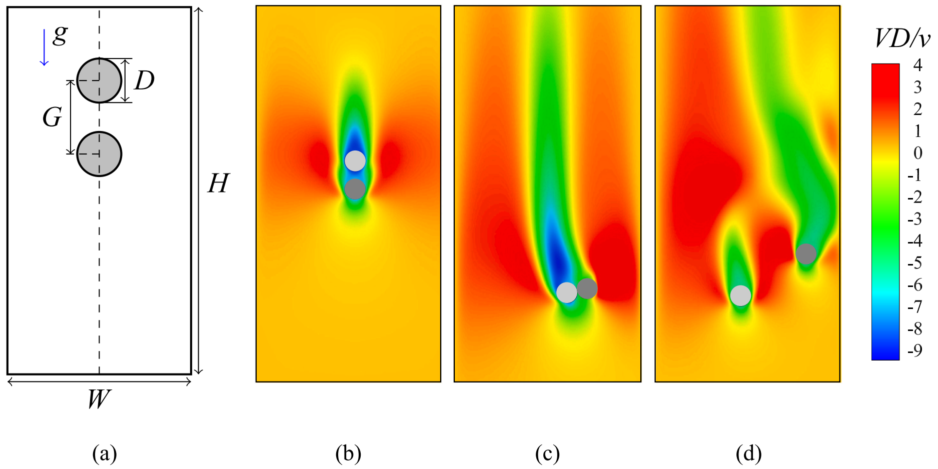
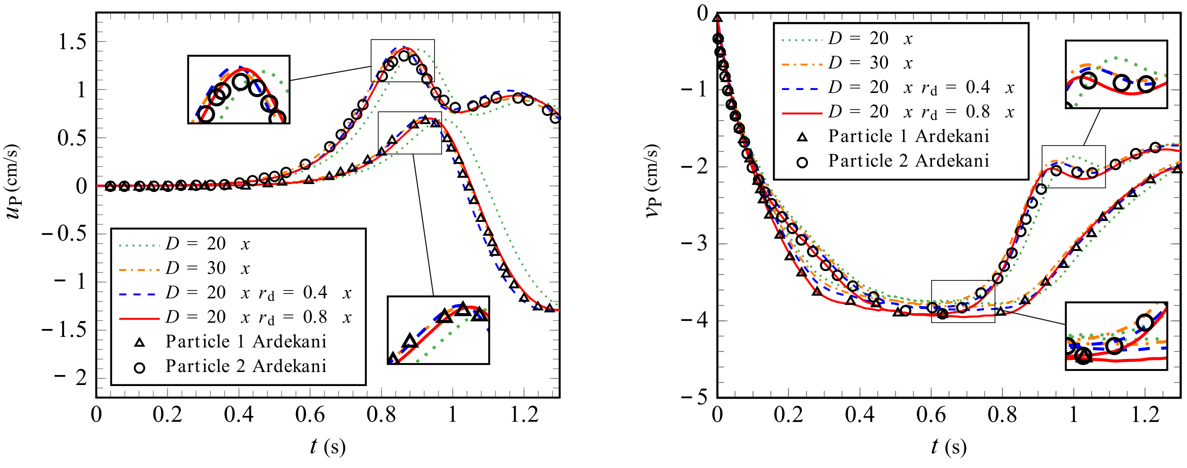


Fig. 14. Sedimentation of a pair of particles in a channel under gravity: (a) computational domain, contours of the vertical velocity V at (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$.



(a)

(b)

Fig. 15. Effect of the interface diffusion on the motion of the two particles in time: (a) the transverse velocity u_p and (b) the longitudinal velocity v_p. The numerical results in Ref. [31] are given for comparison.

CRediT authorship contribution statement

Alessia Abbati: Conceived and designed the analysis, Collected the data, Contributed data or analysis tools, Performed the analysis, Wrote the paper. **Ya Zhang:** Conceived and designed the analysis, Contributed data or analysis tools, Performed the analysis, Wrote the paper. **William Dempster:** Performed the analysis, Wrote the paper. **Yonghao Zhang:** Conceived and designed the analysis, Contributed data or analysis tools, Performed the analysis, Wrote the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Peskin CS. Flow patterns around heart valves: A numerical method. J Comput Phys 1972;10(2):252–71.

[2] Peskin CS. The immersed boundary method. Acta Numer 2002;11:479–517.

[3] Huang W-X, Tian F-B. Recent trends and progress in the immersed boundary method. Proc Inst Mech Eng C 2019;233:7617–36.

[4] Kang SK, Hassan YA. A comparative study of direct-forcing immersed boundary-lattice Boltzmann methods for stationary complex boundaries. Internat J Numer Methods Fluids 2011;66(9):1132–58.

[5] Krüger T, Kusumaatmaja H, Kuzmin A, Shardt O, Silva G, Viggen EM. The lattice Boltzmann method, Vol. 10. Springer; 2017, p. 4–15.

[6] Fadlun E, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J Comput Phys 2000;161(1):35–60.

[7] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. J Comput Phys 2005;209(2):448–76.

[8] Kim W, Choi H. Immersed boundary methods for fluid-structure interaction: A review. Int J Heat Fluid Flow 2019;75:301–9.

[9] Wang Z, Fan J, Luo K. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. Int J Multiph Flow 2008;34(3):283–302.

[10] Wu J, Shu C. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. J Comput Phys 2009;228(6):1963–79.

[11] Breugem WP. A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. J Comput Phys 2012;231(13):4469–98.

[12] Zhou Q, Fan LS. A second-order accurate immersed boundary-lattice Boltzmann method for particle-laden flows. J Comput Phys 2014;268:269–301.

[13] Peng C, Ayala OM, Wang LP. A comparative study of immersed boundary method and interpolated bounce-back scheme for no-slip boundary treatment in the lattice Boltzmann method: Part I, laminar flows. Comput & Fluids 2019;192:104233.

[14] Peng C, Wang LP. Force-amplified, single-sided diffused-interface immersed boundary kernel for correct local velocity gradient computation and accurate no-slip boundary enforcement. Phys Rev E 2020;101(5):053305.

[15] Yu Z, Shao X. A direct-forcing fictitious domain method for particulate flows. J Comput Phys 2007;227(1):292–314.

[16] Bringley TT, Peskin CS. Validation of a simple method for representing spheres and slender bodies in an immersed boundary method for Stokes flow on an unbounded domain. J Comput Phys 2008;227(11):5397–425.

[17] Stockie JM. Modelling and simulation of porous immersed boundaries. Comput Struct 2009;87(11–12):701–9.

[18] Krüger T, Varnik F, Raabe D. Efficient and accurate simulations of deformable particles immersed in a fluid using a combined immersed boundary lattice Boltzmann finite element method. Comput Math Appl 2011;61(12):3485–505.

[19] Maxey M. Simulation methods for particulate flows and concentrated suspensions. Annu Rev Fluid Mech 2017;49:171–93.

[20] Feng ZG, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. J Comput Phys 2004;195(2):602–28.

[21] Suzuki K, Inamuro T. Effect of internal mass in the simulation of a moving body by the immersed boundary method. Comput & Fluids 2011;49(1):173–87.

[22] Ladd AJ. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. J Fluid Mech 1994;271:285–309.

[23] Bouzidi M, Firdaouss M, Lallemand P. Momentum transfer of a Boltzmann-lattice fluid with boundaries. Phys Fluids 2001;13(11):3452–9.

[24] Noble D, Torczynski J. A lattice-Boltzmann method for partially saturated computational cells. Internat J Modern Phys C 1998;9(08):1189–201.

[25] Chen L, Yu Y, Hou G. Sharp-interface immersed boundary lattice Boltzmann method with reduced spurious-pressure oscillations for moving boundaries. Phys Rev E 2013;87(5):053306.

[26] Tsigginos C, Meng J, Gu XJ, Emerson DR. Lattice Boltzmann modeling of fluid-particle interaction based on a two-phase mixture representation. Phys Rev E 2019;100(6):063311.

[27] Zhang Y, Pan G, Zhang Y, Haeri S. A relaxed multi-direct-forcing immersed boundary-cascaded lattice Boltzmann method accelerated on GPU. Comput Phys Comm 2020;248:106980.

[28] Geier M, Greiner A, Korvink JG. Cascaded digital lattice Boltzmann automata for high Reynolds number flow. Phys Rev E 2006;73(6):066705.

[29] Zhang Y, Zhang Y, Pan G, Haeri S. Numerical study of the particle sedimentation in a viscous fluid using a coupled DEM-IB-CLBM approach. J Comput Phys 2018;368:1–20.

[30] Tritton DJ. Experiments on the flow past a circular cylinder at low Reynolds numbers. J Fluid Mech 1959;6(4):547–67.

[31] Ardekani AM, Rangel RH. Numerical investigation of particle - particle and particle - Wall collisions in a viscous fluid. J Fluid Mech 2008;596:437–66.

[32] Premnath KN, Banerjee S. Incorporating forcing terms in cascaded lattice Boltzmann approach by method of central moments. Phys Rev E 2009;80(3):036702.

[33] Ning Y, Premnath KN, Patil DV. Numerical study of the properties of the central moment lattice Boltzmann method. Internat J Numer Methods Fluids 2016;82(2):59–90.

[34] Coreixas C, Chopard B, Latt J. Comprehensive comparison of collision models in the lattice Boltzmann framework: Theoretical investigations. Phys Rev E 2019;100(3):33305.

[35] Guo Z, Zheng C, Shi B. Discrete lattice effects on the forcing term in the lattice Boltzmann method. Phys Rev E 2002;65(4):046308.

[36] Richou AB, Ambari A, Naciri JK. Drag force on a circular cylinder midway between two parallel plates at very low Reynolds numbers - part 1: Poiseuille flow (numerical). Chem Eng Sci 2004;59(15):3215–22.

[37] Dvinsky AS, Popel AS. Motion of a rigid cylinder between parallel plates in stokes flow. part 2: Poiseuille and couette flow. Comput & Fluids 1987;15(4):405–19.

[38] Semin B, Hullin JP, Auradou H. Influence of flow confinement on the drag force on a static cylinder. Phys Fluids 2009;21(10):103604.

[39] Han Y, Cundall PA. Resolution sensitivity of momentum-exchange and immersed boundary methods for solid-fluid interaction in the lattice Boltzmann method. Internat J Numer Methods Fluids 2011;67(3):314–27.

[40] Dvinsky AS, Popel AS. Motion of a rigid cylinder between parallel plates in stokes flow. part 1: Motion in a quiescent fluid and sedimentation. Comput & Fluids 1987;15(4):391–404.

[41] Pianet G, Arquis E. Simulation of particles in fluid: a two-dimensional benchmark for a cylinder settling in a wall-bounded box. Eur J Mech B 2008;27(3):309–21.

[42] Champmartin S, Ambari A. Kinematics of a symmetrically confined cylindrical particle in a "Stokes-type" regime. Phys Fluids 2007;19(7):073303.

[43] Curtis AR, Marr GR. The viscous drag on cylinders falling symmetrically between parallel walls. J Phys D: Appl Phys 1978;11(8):1173–8.

[44] Jayaweera KO, Mason BJ. The behaviour of freely falling cylinders and cones in a viscous fluid. J Fluid Mech 1965;22(4):709–20.

[45] Ghosh S, Stockie JM. Numerical simulations of particle sedimentation using the immersed boundary method. Commun Comput Phys 2015;18(2):380–416.

[46] Ben Richou A, Ambari A, Lebey M, Naciri JK. Drag force on a circular cylinder midway between two parallel plates at Re≪1 part 2: Moving uniformly (numerical and experimental). Chem Eng Sci 2005;60(10):2535–43.

[47] Bampalas N, Graham JM. Flow-induced forces arising during the impact of two circular cylinders. J Fluid Mech 2008;616:205–34.

[48] Fortes AF, Joseph DD, Lundgren TS. Nonlinear mechanics of fluidization of beds of spherical particles. J Fluid Mech 1987;177:467–83.

[49] Aidun CK, Ding EJ. Dynamics of particle sedimentation in a vertical channel: Period-doubling bifurcation and chaotic state. Phys Fluids 2003;15(6):1612–21.

[50] Wang L, Guo ZL, Mi JC. Drafting, kissing and tumbling process of two particles with different sizes. Comput & Fluids 2014;96:20–34.

[51] Zhang P, Galindo-Torres SA, Tang H, Jin G, Scheuermann A, Li L. An efficient discrete element lattice Boltzmann model for simulation of particle-fluid, particle-particle interactions. Comput & Fluids 2017;147:63–71.

[52] Niu XD, Shu C, Chew YT, Peng Y. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. Phys Lett Sect A: General Atomic Solid State Phys 2006;354(3):173–82.

[53] Zhang H, Tan Y, Shu S, Niu X, Trias FX, Yang D, Li H, Sheng Y. Numerical investigation on the role of discrete element method in combined LBM-IBM-DEM modeling. Comput & Fluids 2014;94:37–48.