A hybrid LBM-DEM numerical approach with an improved immersed moving boundary method for complex particle-liquid flows involving adhesive particles

Wenwei Liu, Chuan-Yu Wu*

Department of Chemical and Process Engineering, University of Surrey, Guildford, GU2 7XH, UK

Abstract
This paper presents a hybrid numerical framework for modelling solid-liquid flow with particle adhesion based on a coupled single-relaxation-time lattice Boltzmann method (LBM) and a discrete element method (DEM) for adhesive particles. The LBM is implemented with an improved immersed moving boundary (IMB) method for solving complex fluid-particle interactions using a fast linear approximation of partially intersected volume between a particle and a lattice cell. It is shown that the linear approximation of the solid fraction is very efficient with good accuracy. In order to model particle adhesion, the Johnson-Kendall-Roberts (JKR) adhesive contact mechanics is adopted with a novel implicit solution to obtain the particle-particle normal force in the DEM. The developed hybrid IMB-LBM-DEM numerical framework is verified with several benchmark tests. The results confirm that the hybrid numerical approach is capable of deriving more detailed flow field between dense particles with relatively low grid resolution, as well as effectively capturing the adhesive mechanics between microspheres.

Keywords: Lattice Boltzmann method; Discrete element method; Immersed boundary method; Adhesion; JKR model

*Corresponding author:
Prof. Chuan-Yu Wu, Email: c.y.wu@surrey.ac.uk
1. Introduction

The transport of solid particles with fluids ubiquitously exists in a variety of scientific issues and engineering applications. In these problems, the fluid-particle interactions are of great significance to well describe the mechanics of individual particles, which are crucial to understanding the behaviour of the system. Because of its complex nature, it is still challenging to accurately characterise the fluid-particle hydrodynamic interactions. Nevertheless, a few numerical techniques available analyse the complex fluid-particle multiphase flows. For example, the computational fluid dynamics (CFD) coupled with the discrete element method (DEM) is the most widely used numerical approach [1-7] [Hu92, Feng94a, Feng94b, Kuipers92, Tsuji93, Kafui02, Zhu07]. In this method, the fluid phase is solved using the finite volume method (FVM) or the finite element method (FEM) based on the discretization of the macroscopic Navier-Stokes equation, while the particle phase is solved with the Newton’s second law. The coupling between CFD and DEM is realised using an interaction force term based on the local porosity of the particles in a computational fluid cell. However, the drawback of this approach lies in that the mesh grid size has to be larger than the particle size, which results in less details at the particle scale. Also, the finite element discretization becomes more intractable when simulating dense particle-fluid flows, due to the transient nature of contacts and the tortuosity of the void spaces.

In the recent two decades, lattice Boltzmann method (LBM) has become a promising alternative approach for solving fluid flows [8-10] [Chen92, Chen98, Gladrow04]. In LBM, the fluid domain is discretized with a regular orthogonal grid, which is called lattice and is similar to the mesh grid in CFD. The fluid in each lattice is represented by packets of fictitious
‘particles’, of which the motions follow the lattice Boltzmann equation. During every computational time step, the fluid flow in each lattice is updated only with its local information, which makes it naturally suitable for parallel computing. For the fluid-particle interactions in LBM, several coupling techniques were proposed, including the modified bounce-back method (MBB) [11-12] [Ladd94ab], the immersed boundary method (IBM) [13-15] [Peskin77, Peskin02, Feng03], and the immersed moving boundary (IMB) method [16] [Noble98].

The MBB uses a discretized lattice representation of solid particles. Therefore, it can be readily used to describe any particle shape. The bounce back is assumed to occur at the boundary between solid lattice nodes and fluid lattice nodes [11] [Ladd94a]. The main drawback of the MBB is the stepwise lattice representation, which can result in the fluctuation of the hydrodynamic interactions [17] [Feng07]. Only by increasing the grid resolution (i.e. the particle-to-lattice size ratio) can one obtain a smoother and more accurate result, but this significantly increases the computational cost.

The IBM was proposed by Peskin [13-14] [Peskin77, Peskin02] and was widely applied in the conventional CFD-DEM models [18] [Mittal05]. In this method, the fluid and the particles are represented with fixed Eulerian mesh grids and the moving Lagrangian nodes, respectively. The particles are assumed to be deformable with a large stiffness. The effect of the immersed particle boundary on the fluid are first modelled by restoring forces, which tend to keep the particle to its original shape, on the Lagrangian grid based on the no-slip boundary condition. Then the restoring forces are distributed to their surrounding Eulerian grids and considered in the governing equations as the external force terms. The drawback of the IBM lies in the extra cost of the data storage for the Lagrangian grid of the moving particles, which
becomes less efficient when the particle concentration is high. Moreover, the interaction between the two coordinate systems is not compatible to the parallel nature of LBM, which requires a more complicated algorithm.

Compared with the above method, the IMB method proposed by Noble and Torczynski [16] [Noble98] is easier to implement and provides smoother results, and has been widely applied in the numerical study of dense fluid-particle flows [17, 19-22] [Feng07, Strack07, Feng10, Owen11, Han12]. The basic idea of the IMB method is the introduction of an additional collision operator to the conventional lattice Boltzmann equation, which is modified with a weighting function that depends on the solid fraction in the local lattice cell. Thus, the only challenge in this method is the accurate estimation of the solid fraction in each lattice cell. However, there were some issues that were not properly addressed in the original IMB method. For instance, when multiple particles intersect with the same lattice cell, the contributions to the total weighting function from every intersecting particle are not considered in a self-consistent way, which can cause the divergency in the computation. Furthermore, the hydrodynamic interactions on each solid particle in the same cell should also be weighted based on its corresponding solid fraction contribution. Such discrepancies is generally negligible in the dilute particle-fluid flows, as interparticle contacts barely occur in this case. Nevertheless, for the dense multiphase flow where numerous close contacts are developed, it could lead to severe numerical problem, such as divergency and irrational particle collisions.

Therefore, in this paper, we present an enhanced hybrid numerical framework for modelling particle-fluid flows with particle adhesion based on a coupled single-relaxation-time LBM and DEM. An improved IMB method is developed for solving the fluid-particle
interactions, where the solid fraction is estimated with a new linear approximation method with high efficiency and accuracy. In addition, the Johnson-Kendall-Roberts (JKR) adhesive contact mechanics is adopted in the DEM to describe the interparticle normal force for adhesive particles. To the best of our knowledge, there were very limited LBM-DEM studies on two-phase flows with adhesive particles, despite the ubiquity of their application in almost all areas of engineering, biology, agriculture and physical sciences [23-29] [Li11, Marshall14, Liu15, Chen16a, Chen16b, Dominik97, Kinch07]. The challenge lies in coupling adhesion, elastic contact forces and frictional forces in the short-range particle-particle interaction zone as well as the coupling with the fluid forces (e.g. buoyancy, drag and lubrication) across both the long-range length scale and the time scale. In this work, an implicit solution is employed to obtain the particle-particle adhesive normal force. The hybrid IMB-LBM-DEM numerical framework is then validated with several benchmark tests. The paper is organised as follows. The numerical model is given in Section 2 in detail, where a brief review of the LBM, the improved IMB method, the fast linear approximation method to estimate the solid fraction, as well as the DEM for adhesive particles, are described. Section 3 presents the model validations and conclusions are drawn in Section 4.

2. Numerical model

2.1 Lattice Boltzmann method

LBM was originally developed from the lattice gas automata (LGA) method, which can be considered as a simplified fictitious molecular dynamics model to simulate fluid flows [9,
In LBM, the fluid domain is discretized with a number of lattices. In each lattice, the packets of fluid are described by density distribution functions $f_i(x,t)$, which relate the probable amount of fictitious fluid ‘particles’ moving at a discrete speed in a discrete direction at each lattice node within each time increment. In every lattice, the density distribution functions follow the same discretized speed model, for which various algorithms, such as D2Q7, D2Q9 models for 2D and D3Q15, D3Q19, D3Q27 models for 3D, are proposed. In our numerical framework, the widely used 3D discretization schemes D3Q19 model is adopted, in which the fluid particles at each lattice are allowed to move to its 26 immediate neighbours with 18 different velocities, $e_i (i=1-18)$, as shown in Fig. 1.

The density distribution functions $f_i(x,t)$ are governed by the lattice Boltzmann equation (LBE) [9] [Chen98].
\[
f_i(x + e_i \Delta t, t + \Delta t) = f_i(x, t) + \Omega_i[f_i(x, t)] ,
\]

(1)

where the vector \( x \) denotes the position of the node for which the calculation is being carried out, \( \Delta t \) is the time step, and \( \Omega_i[f_i(x, t)] \) is the collision operator that controls the relaxation rate of the density distribution functions \( f_i(x, t) \) that meet at a node.

At each node, Eq. (1) is evolved by the collision and streaming processes at each time step. Collision (also known as relaxation) redistributes the functions that arrive at each node and then streaming (also known as convection) propagates the redistributed functions to their nearest neighbour nodes. Over a number of time steps the ordering of the streaming and collision operations is irrelevant. For the collision process, a single-relaxation-time LBE linearized by the Bhatnagar-Gross-Krook approximation is employed [8,10,30-31] [Bhatnagar54, Chen92, Qian92, Wolf04], where the collision operator reads

\[
\Omega_i = -\frac{\Delta t}{\tau}[f_i(x, t) - f_i^{eq}(x, t)] + F_i \Delta t .
\]

(2)

Here, \( \tau \) is the dimensionless relaxation parameter, \( F_i \Delta t \) represents a body force acting on the fluid and \( f_i^{eq}(x, t) \) is the equilibrium distribution function defined as

\[
f_i^{eq}(x, t) = \rho \omega_i [1 + \frac{e_i \cdot u}{c_s^2} + \frac{(e_i \cdot u)^2}{2c_i^2} - \frac{u^2}{2c_i^2}] .
\]

(3)

In Eq. (3), \( \omega_i \) is the weight coefficient based on the lattice speed model. For the D3Q19 model, \( \omega_0 = 1/3, \omega_{1-6} = 1/18, \omega_{7-18} = 1/36 \). \( c_s = c / \sqrt{3} \) is the lattice sound speed, where \( c = |\Delta x / \Delta t| \) is...
the lattice speed. $\rho$ and $\mathbf{u}$ are the macroscopic fluid density and velocity, which are determined as

$$\rho = \sum_i f_i,$$
$$\rho \mathbf{u} = \sum_i f_i \mathbf{e}_i,$$  \hspace{1cm} (4)

while the fluid pressure field $p$ is determined by $p = c_s^2 \rho$. To recover the incompressible Navier-Stokes equations, the kinematic viscosity of the fluid $\nu$ is not directly used in the LBM model but implicitly determined by the discretization and numerical parameters as follows

$$\nu = \frac{1}{3} \left( \frac{\tau}{2} \right) \frac{(\Delta x)^2}{\Delta t}.$$  \hspace{1cm} (5)

The body force $F_i \Delta t$ in Eq. (2) is given in the following form, in order to correctly recover the Navier-Stokes equation with a body force term, according to reference [32] [Guo02],

$$F_i = (1 - \frac{1}{2 \tau}) \omega_i \left[ \frac{\mathbf{e}_i - \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})}{c_s^4} \mathbf{e}_i \right] \cdot \mathbf{F},$$  \hspace{1cm} (6)

where $\mathbf{F}$ is macroscopic body force. Correspondingly, the macroscopic fluid velocity in Eq. (4) should be modified as

$$\rho \mathbf{u} = \sum_i f_i \mathbf{e}_i + \frac{\Delta t}{2} \mathbf{F}.$$  \hspace{1cm} (7)
2.2 Boundary conditions

In LBM, the boundary conditions are implemented using the density distribution functions. In the present work, the ‘no-slip’ wall boundary conditions, the periodic boundary conditions and the moving wall boundary conditions are considered. The ‘no-slip’ boundary condition at the interface between the fluid and the stationary solid wall can be imposed by the so-called bounce-back rule [11] [Ladd94a], which is simply reversing the incoming density distribution functions from the fluid node back to the directions from which they come at all wall boundary nodes. The bounce-back rules can be defined as

\[ f_{-i}(\mathbf{x}, t + \Delta t) = f_{i}^+(\mathbf{x}, t) , \]

(8)

where \(-i\) denotes the opposite direction to \(i\) and \(f_i^+\) represents the post-collision density distribution function. Note that the second-order accuracy can be achieved when the collision and streaming processes are also carried out at the solid boundary nodes [33] [He97]. Periodic boundary conditions are implemented in LBM in such a way that the density distribution functions exiting the domain at one end are duplicated to a virtual node at the other end. Then a normal streaming process takes place between the virtual node and the corresponding nodes at the other end. It should be noted that, in our LBM, the solid particles immersed in the fluid are treated as moving wall boundaries. The interactions between the moving particles and the fluid are very crucial in the coupling of LBM and DEM. The moving wall boundary conditions
are hence discussed separately in the next section.

2.3 Immersed moving boundary method

For moving boundaries between the solid particles and the fluid, a physically correct ‘no-slip’ boundary condition is required to impose at the interface, for which we must first identify the solid particle’s boundary by representing the particle with the lattice nodes. Figure 2 illustrates the lattice discretization of a circular particle, where the 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. Obviously, the stepwise lattice representation of the surface of a circular particle is neither accurate nor smooth unless a sufficiently small lattice spacing is used.
Fig. 2 Lattice representation of a circular solid particle showing solid boundary nodes (orange), fluid boundary nodes (green) and internal solid nodes (blue). The red and purple parts represent the solid coverage ratio at a solid boundary node and a fluid boundary node, respectively.

Based on the lattice representation, a modified bounce-back (MBB) method was developed by Ladd [11] to model the fluid-particle interaction with moving boundaries. This method assumes that the bounce-back occurs at the solid-fluid interface at halfway between the fluid boundary nodes and solid boundary nodes. Density functions at the corresponding fluid/solid boundary nodes are updated based on the consideration of the solid particle’s velocity at the boundary link. The total hydrodynamic force exerted on the solid particle is calculated by summing up the net momentum exchange over all the boundary links. The details of the MBB method can be found in many previous literature, such as [11,17,19,21] [Ladd94a, Feng07, Strack07, Owen11], which are not repeated here. However, two significant drawbacks of the MBB method are the disparity between the physical and simulated boundary shape, and the occurrence of fluctuations in the induced hydrodynamic force and torque [21] [Owen11].

As an improvement, an immersed moving boundary (IMB) technique for LBM-DEM coupling was proposed by Noble and Torczynski [16] [Noble98], which is also known as the partially solid scheme. Two objectives of the method were to overcome the momentum discontinuity of MBB-based techniques and to provide adequate representation of non-conforming boundaries at lower grid resolutions. It was also important to retain two critical advantages of the LBM, namely the locality of the collision operator and the simple linear
streaming operator, and thus facilitate solutions involving large numbers of irregular shaped moving boundaries. In this method the lattice Boltzmann equation is modified to include a term that is dependent on the solid coverage ratio of the cell (see Fig. 2), thus improving the boundary representation and smoothing the hydrodynamic forces calculated at a particle’s boundary nodes as it moves relative to the grid.

In the IMB method, the modified lattice Boltzmann equation can be written to include the body force term as

\[ f_i(x + e_i \Delta t, t + \Delta t) = f_i(x, t) - (1 - B_n) \left[ \frac{A}{\tau} (f_i(x, t) - f_i^{eq}(x, t)) \right] + B_n \Omega_i + (1 - B_n) F_i \Delta t, \]  

(9)

where \( \Omega_i \) is the additional collision term, and \( B_n \) is a weighting function based on the total solid fraction in each cell, \( \varepsilon_n \). Note that in the original method, both the solid fraction and the weighting function are summations of all coverage contributions from solid particles that intersect the same cell so that

\[
\varepsilon_n = \sum_s \varepsilon_s, \\
B_n = \sum_s B_s, 
\]

(10)

where \( \varepsilon_s \) and \( B_s \) are the contributions from each solid particle. The additional collision term modifies the momenta of mapped particle nodes and accounts for fluid interaction with any solid particles present in the cell. The weighting function is suggested to have the following form

\[
B_s(\varepsilon_s, \tau) = \frac{\varepsilon_s (\tau / \Delta t - 0.5)}{(1 - \varepsilon_s) + (\tau / \Delta t - 0.5)}. 
\]

(11)
As the value of the solid fraction $\varepsilon_s$ varies from 0 (a completely fluid cell) to 1 (a completely solid cell), $B_s$ also varies from 0 to 1. Eq. (11) returns the original collision equation for pure fluid when $B_s=0$, and returns the new collision operator $\Omega^s_i$ plus the distribution from the previous time step when $B_s=1$. The new collision operator is given by

$$\Omega^s_i = f_{s,i}(\mathbf{x},t) - f_{out}^{\rho}(\rho,\mathbf{u}) + f_{out}^{\rho}(\rho,\mathbf{u}_s) - f_i(\mathbf{x},t),$$  \hspace{1cm} (12)$$

where $\mathbf{u}_s$ is the velocity of the solid particle. Note the straightforward implementation of the method: a single term ($\Omega^s_i$) is added to the LBE and two coefficients ($1-B_n$) in the equation are modified. Only quantities already available on the mesh or easily derived are used. No additional data storage or organization is needed, which is a crucial issue in most moving boundary formulations. Calculations of the standard LBE for lattice nodes that are partially or completely covered by solids are simply replaced by the modified Eq. (9). The total hydrodynamic force and torque acting on the solid particle is determined by summing the change of momenta due to the additional collision operator over all lattice directions at each node and then over all fluid boundary, solid boundary and internal solid nodes, which are expressed as

$$\mathbf{F}_j = -\sum_n B_n (\sum_i \Omega^s_i \mathbf{e}_i),$$

$$\mathbf{T}_j = -\sum_n [(\mathbf{x}_n - \mathbf{X}_p) \times B_n (\sum_i \Omega^s_i \mathbf{e}_i)].$$ \hspace{1cm} (13)
Here, $\mathbf{x}_n-\mathbf{X}_p$ is the vector from the center of rotation to the coupled node and the minus sign represents the direction of the force and torque according to the Newton’s third law.

However, the original method becomes inaccurate when more than one particle intersects with the same cell. As $B_s$ is non-linearly dependent on $\varepsilon_s$ (see Eq. (11)), the sum of $B_s$ does not equal to that calculated using the sum of solid fraction $\varepsilon_s$, i.e. $B_n = \sum B_s(\varepsilon_s, \tau) \neq B_s(\sum \varepsilon_s, \tau)$, which will lead to the incorrect weighting function in the solid-fluid coupling term. For example, consider that two particles intersect with the same cell with solid fractions of $\varepsilon_{s,1}=0.4$ and $\varepsilon_{s,2}=0.6$, respectively. The total solid fraction in this cell is $\varepsilon_n=\varepsilon_{s,1}+\varepsilon_{s,2}=1$, indicating a completely solid cell. Then the corresponding total weighting function $B_n$ should be physically equal to one. Nevertheless, according to Eq. (11), the total weighting function is only $B_n=0.18, 0.39, 0.52$, given that $\tau/\Delta t=0.6, 0.8, 1.0$, respectively. Figure 3(a) shows the variations of the weighting function $B_s$ with the solid fraction of a single particle $\varepsilon_s$. We can see that $B_s$ increases non-linearly with $\varepsilon_s$. Only with a relatively large relaxation parameter, the relationship between $B_s$ and $\varepsilon_s$ approaches linear. Figure 3(b) further presents a comparison between the two different ways of calculating the total weighting function using Eq. (10) and Eq. (11), for the case that two particles intersect with the same cell, i.e. $B_n = B_s(\varepsilon_s, \tau) + B_s(\varepsilon_n - \varepsilon_s, \tau)$ and $B_n = B_s(\varepsilon_n, \tau)$. Three different total solid fractions are used for the calculation, $\varepsilon_n=0.6, 0.8, 1.0$, and the relaxation parameter is fixed at $\tau=0.8$. It can be seen that the original IMB method always underestimates the total weighting function. This distinction is substantially caused by the linear inconsistency in the definition of $B_n$ and $B_s$. Moreover, in the force calculation, the contribution to the force on each particle should also be weighted based on its solid fraction, which is actually not fully considered in the original IMB method. Therefore, an improved
IMB method is proposed in the current paper.

The modified IMB-LBE is rewritten as

\[ f_i(x + e_i \Delta t, t + \Delta t) = f_i(x, t) - (1 - B_n) \left[ \frac{\Delta t}{\tau} \left( f_i(x, t) - f_i^{eq}(x, t) \right) \right] + \sum \Omega_i^t + (1 - B_n) F_i \Delta t, \]  

(14)

where the term \( B_n \Omega_i^s \) is replaced with the sum of the contribution from each particle that intersects with the cell. \( B_n \) keeps the same form as that in the original method, but \( B_s \) is modified as

\[ B_s(\varepsilon_s, \tau) = \frac{\varepsilon_s (\tau / \Delta t - 0.5)}{(1 - \varepsilon_n) + (\tau / \Delta t - 0.5)}, \]  

(15)

where \( \varepsilon_n \) is still the total solid fraction in the same cell. In this case, when there is only one particle that intersects with a cell, Eqs. (14) and (15) become exactly the same as the original method. When there is more than one particle intersecting with a cell, their contributions to the total weighting function \( B_n \) can be linearly summed up, which ensures that \( \sum B_s(\varepsilon_s, \tau) = B_s(\sum \varepsilon_s, \tau) \). The additional collision term \( \Omega_i^s \) also keeps the same form as in Eq. (12) but is updated with the velocity of corresponding solid particle. Similarly, the equations used to compute the force and torque are modified as

\[ F_f = -\sum_n B_s(\sum \Omega_i^t e_i), \]

\[ T_f = -\sum_n [(x_n - X_p) \times B_s(\sum \Omega_i^t e_i)], \]  

(16)

where the total weighting function \( B_n \) is replaced by the weighting function \( B_s \) of the corresponding particle in the cell. These modifications enable the correct coverage physics in the cell with multiple particles to be properly described, although this circumstance does not
occur often when the particle-lattice size ratio is relatively large in the LBM-DEM simulation. Nevertheless, it becomes important for the dense particle-liquid flows, where particles are in close contact with each other.

Fig. 3 (a) Weighting function $B_s(\varepsilon_s, \tau)$ as a function of the solid fraction $\varepsilon_s$ for different relaxation parameters $\tau$. (b) The total weighting function $B_n$ as a function of the solid fraction of a particle for the case that two particles intersect with the same cell. Comparison is made between the modified and the original weighting function $B_s$ for different total solid fractions $\varepsilon_n$. The relaxation parameter is fixed at $\tau=0.8$.

2.4 An accurate estimation of solid fraction in the lattice cell

In the immersed moving boundary method described above, it is clear that accurate calculation of the solid fraction in each lattice cell is crucial. In this section, a fast linear approximation method is introduced, which was recently proposed by Jones and Williams [34] [Jones17], to estimate the solid fraction of a single particle that intersects with a lattice cell. Figure 4 shows the schematic of the linear approximation in a simplified 2D view for better
illustration. In this approach, the solid fraction is simply approximated by a linear function,

\[ \varepsilon_s = -D + f(r), \quad (17) \]

where \( D \) is the distance from the cell center to the particle surface. \( f(r) \) is a function of the normalized particle radius. Eq. (17) is derived from the analytical solution to the intersection volume calculation for a specific cell orientation with respect to the particle surface, based upon the following assumptions:

(i) The cell and particle centers lie along a single axis, i.e. they have a \( z \)-component of zero.

(ii) A pair of opposing faces in the cell is parallel to the plane tangential to the sphere surface.

(iii) The sphere surface intersects only the remaining cell faces, and not those parallel to the tangent plane.

As illustrated in Fig. 4, by setting the particle center as the origin, the solid fraction equals to the intersection volume \( V_i \), i.e.

\[ \varepsilon_s = V_i = V_a - V_b, \quad (18) \]

where \( V_b \) is the volume of the cuboid between the lower cell face and the origin, \( V_b = y_c - 0.5 \). \( y_c \) is the \( y \)-coordinate of the cell center. \( V_a \) is calculated with the integral,
\[ V_a = \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} (r^2 - x^2 - y^2) \, dx \, dy \]
\[ = \left( \frac{1}{12} - r^2 \right) \tan^{-1} \left( \frac{0.5 \sqrt{r^2 - 0.5}}{0.5 - r^2} \right) + \frac{1}{3} \sqrt{r^2 - 0.5}, \]  \hspace{1cm} (19)
\[ + (r^2 - \frac{1}{12}) \tan^{-1} \left( \frac{0.5}{\sqrt{r^2 - 0.5}} \right) - \frac{4}{3} r^3 \tan^{-1} \left( \frac{0.25}{r \sqrt{r^2 - 0.5}} \right) \]

which is constant value for a fixed particle radius. Note that \( y_c = r + D \), then by substituting \( V_a \) and \( V_b \) into Eq. (18), we have

\[ f(r) = V_a - r + 0.5. \]  \hspace{1cm} (20)

This approach is also valid in 2D, where the integral to compute \( V_a \) becomes

\[ V_a = \int_{-0.5}^{0.5} (r^2 - x^2) \, dx = 0.5 \sqrt{r^2 - 0.25} + r^2 \tan^{-1} \left( \frac{0.5}{\sqrt{r^2 - 0.25}} \right). \]  \hspace{1cm} (21)

This linear approximation is proved to be surprisingly accurate when applied in the general case and significantly saves the computational cost [34] [Jones17]. However, according to the assumptions, some special cases still need to be handled. For instance, if the cell only intersects with the particle at a corner, Eq. (17) could give negative values due to the dissatisfaction of the assumptions. Therefore, in the actual implementation, we still use the simple Eq. (17) to calculate the solid fraction in any case but force the solid fraction to be in the range of \([0,1]\). That is, \( \varepsilon_s = 0 \) if Eq. (17) gives negative values; \( \varepsilon_s = 1 \) if Eq. (17) gives values above one. Furthermore, it is obvious that the solid fractions of the cells completely inside and outside the particle are one and zero, respectively. Therefore, a further speed up of the algorithm can be
achieved by a simple search with the distance from the cell center to the particle center. Specifically, in this current work, only the lattice cells with the center-to-center distance in the range \([r-h, r+h]\) are computed with the linear approximation method, which are contained in a spherical shell with the thickness of \(2h\) lattice units, with \(h\) being a tunable variable. Usually, \(h\) can be set to one as the particle diameter is at least 5 lattice units in the LBM-DEM. Table 1 summarises the algorithm of the calculation of the solid fraction.

![Diagram](image)

**Fig. 4 Schematic of linear approximation in 2D.**

**Table 1 Algorithm of the calculation of solid fraction in a lattice cell**

| Step | Action |
|------|--------|
| **DO n=1, Num_of_Particle** (loop over all the solid particles) | |
| **Step 1** | Find all the lattices that fully contain the current solid particle. |
| **DO m=1, Num_of_Lattice** (loop over all the lattices found in Step 1) | |
Step 2  Calculate the distance between the lattice cell and center of particle.

Step 3  IF the distance ≥ particle radius + h

          Solid fraction = 1.0

ELSE IF the distance ≤ particle radius – h

          Solid fraction = 0.0

ELSE IF the distance is in between

          Use Eqs. (17), (19) and (20) to calculate the solid fraction.

          END IF

Step 4  Modify the unreasonable values of solid fraction calculated by the linear approximation equations.

          IF solid fraction ≤ 0.0, THEN solid fraction = 0.0

          IF solid fraction ≥ 1.0, THEN solid fraction = 1.0

Step 5  Calculate the total solid fraction in the current cell and modify the unreasonable values of total solid fraction.

          IF total solid fraction ≥ 1.0, THEN total solid fraction = 1.0

Step 6  Record the particle ID that intersects with the cell.

          END DO (move on to next lattice cell)

          END DO (move on to next solid particle)

To assess the accuracy and efficiency of the linear approximation method, we calculate the solid fraction of a single sphere with increasing particle-to-lattice size resolutions and compare the results with other methods, including the Monte Carlo method and the polyhedron
approximation method [34] [Jones17]. The Monte Carlo method is a statistical sampling technique, which randomly places a number of sampling points with uniform distribution in the lattice cell. The distance between each sample point and the sphere center is calculated and compared with the sphere radius. If the distance is smaller than the radius, the corresponding sample point is counted as an inside point. Then the intersecting solid fraction is estimated as the ratio of the number of the inside points to the total number of sample points. Obviously, the more sample points are placed, the more accurate the estimation is, but the longer time the computation takes. The polyhedron approximation is a simplified analytical computational method, in which the intersecting solid fraction is approximated as a combination of polyhedra in 3D or a polygon in 2D. The key steps in this method is identifying the polyhedron that is defined by the intersecting points at the edges of the lattice cell. Generally, the larger the sphere-to-lattice size ratio is, the more accurate the approximation is, because the volume of the unaccounted spherical cap in the approximation becomes smaller.

For the test setup, we place a sphere in the center of a cuboid box with the size of \((d+2)^3\), where \(d\) is the sphere diameter in lattice unit and increases from 10 to 100, corresponding to the total number of lattice cells from \(10^3\) to \(10^6\). The computation time is estimated on a PC with the configuration of Intel Core i7-6700, 4 cores (8 threads), 3.4 GHz and 16 GB RAM, and without any parallel computing. Upon the calculation of the solid fraction in every lattice cell, the relative error in the total volume of the sphere can be estimated. Figure 5 shows the comparison of the computational performance for different methods. It is clear that the computational speed of the Monte Carlo method is of the order \(O(n)\) and \(O(d^3)\), where \(n\) denotes the total number of the sample points, while the relative error decreases less than 3 orders of
magnitude as the resolution increases by an order of 1. Besides, the accuracy of MC seems to be saturated when \( n \geq 1,000 \). For the other two methods, the computational speed is slightly above the order of \( O(d) \), which is much faster than the MC. However, the relative error of the linear approximation is generally an order of magnitude smaller than that of the polyhedron approximation. Although the relative error is still 1 order higher than the MC, the overall performance of the linear approximation is very satisfactory, considering the fast computational speed and the acceptable relative error. A more detailed comparison of the computational performance of the linear approximation with other available methods, including the full analytical solution method, the cell decomposition method, and the edge-intersection averaging method, can be found in [34] [Jones17].
2.3 The discrete element method for adhesive particles

In DEM, the particle’s motion is described by the Newton’s second law [23-24] [Li11, Marshall14], i.e.
\[
\frac{dU_p}{dt} = F_f + F_c + G, \\
I \frac{d\Omega_p}{dt} = M_f + M_c,
\]

where \( U_p \) and \( \Omega_p \) are, respectively, transitional velocity and rotational velocity of an individual particle. \( m \) is the particle mass and \( I \) is the moment of inertia. \( G \) is the gravity. \( F_f \) and \( M_f \) denote the fluid force and torque acting on each individual particle, respectively, and \( F_c \) and \( M_c \) are the force and torque resulted from the interparticle contact, respectively. The contact force and torque can be decomposed as

\[
F_c = F_n n + F_s t, \\
M_c = r_p F_s (n \times t) + M_r (t_s \times n) + M_t n,
\]

where \( F_n \) is the normal force including the elastic contact force and the damping force, \( F_s \) is the tangential force due to the sliding friction, \( M_r \) is the rolling resistance and \( M_t \) is the twisting resistance. \( r_p \) is the particle radius. \( n, t_s, t_r \) and \( t_c \) are the normal, tangential and rolling direction unit vectors, respectively.

In a collision process between two adhesive microparticles, the particles undergo the jump-on and pull-off processes when they contact with and detach from each other, respectively, which greatly differ from the collision of granular particles [24] [Marshall 14]. As shown in Fig. 6, at the jump-on point (point A), the contact region area suddenly goes from zero to a finite value and the contact force suddenly goes from zero to a negative value, which leads to a first-contact energy loss. The necking behavior of the material when the particles are pulled away
from each other allows the adhesive force to act even when the normal overlap becomes negative, resulting in another an energy loss. The particles will finally detach from each other when a critical pull-off force is reached (point C). The energy dissipation due to the jump-on and pull-off behaviour of the adhesive contact is estimated as \[ \Delta E_{ad} = \int_{-\delta_c}^{0} F_{ad} d\delta_N = 22.51\left(\frac{\gamma_s R^4}{E^2}\right)^{1/3}. \] (24)

![Force-displacement curve during the normal loading and unloading of two adhesive spheres.](image)

In Eqs. (24) and (25), \( F_{ad} \) and \( F_{nd} \) represent the adhesive force and damping force in the normal
direction between two microparticles, respectively. $\delta_N$ is the normal overlap which equals $\delta_C$ at the critical pull-off point, and $\gamma$ is the surface energy. $R$ is the effective radius and $E$ is defined as the effective elastic modulus between two contacting particles,

$$\frac{1}{R} = \frac{1}{r_{p,i}} + \frac{1}{r_{p,j}},$$

$$\frac{1}{E} \equiv \frac{1 - \sigma_i^2}{E_{p,i}} + \frac{1 - \sigma_j^2}{E_{p,j}},$$

(26)

where $r_p$, $E_p$ and $\sigma$ denote the particle radius, elastic modulus and Poisson’s ratio, respectively, and the subscripts $i$ and $j$ correspond to the two particle indexes. Note that we do not consider the plastic deformation in the present work, because it only becomes non-negligible when the impact velocity of particles is much larger than their critical sticking/rebound velocity [23,36] [Chen15, Li11]. Combing these two energy dissipation mechanisms, the instantaneous normal contact force $F_n$ is given by

$$F_n = F_{nd} + F_{md} = 4F_C \left[ \left( \frac{a}{a_0} \right)^3 - \left( \frac{a}{a_0} \right)^{3/2} \right] + \eta_N \mathbf{v}_n \cdot \mathbf{n},$$

(27)

where the adhesively normal contact force is described by the JKR (Johnson-Kendall-Roberts) model [37] [Johnson71] and the damping force is assumed to be proportional to the rate of change of material deformation [23-24] [Li11, Marshall14]. $F_C$ is the critical pull-off force derived from the JKR theory, $F_C=3\pi\gamma R$ [37] [Johnson71], and $a$ is the radius of the contact area with $a_0$ being the equilibrium contact radius, which is given as
\[ a_0 = \sqrt{\frac{9\pi\gamma R^2}{E}}. \]  

\( \eta_N \) is the normal dissipation coefficient, and \( v_R \) is the relative velocity at the contact point on particle surfaces. To minimize the computational time, \( F_{ad} \) and \( a_0 \) as functions of \( \delta_N / \delta_C \) are pre-computed at the beginning, and then we use a look-up table to determine \( F_{ad} \) and \( a(t) \) for the given value of \( \delta_N \) at each time step.

Apart from the normal deformation, the interparticle sliding, twisting and rolling frictions are also considered and approximated with a linear spring-dashpot-slider model. They can be expressed as

\[
F_s = -\min\left[k_T\left(\int_0^t v_R(\tau) \cdot t_s \, d\tau\right) + \eta_T v_R \cdot t_s, F_{s,\text{crit}}\right],
\]
\[
M_t = -\min\left[k_T a^2\left(\int_0^t \Omega_t(\tau) \, d\tau\right) + \eta_T a^2, M_{t,\text{crit}}\right],
\]
\[
M_r = -\min\left[4F_C \left(\frac{a}{a_0}\right)^{3/2}\left(\int_0^t v_L(\tau) \, d\tau\right), M_{r,\text{crit}}\right],
\]

where \( v_R \cdot t_s, \Omega_t \) and \( v_L \) stand for the relative sliding, twisting and rolling velocity, respectively. \( k_T \) and \( \eta_T \) are the tangential stiffness and dissipation coefficient, respectively. According to Eq. (29), the sliding, twisting and rolling resistances first increase cumulatively with the increase of the corresponding displacements. Once reaching certain critical values, i.e. \( F_{s,\text{crit}}, M_{t,\text{crit}} \) and \( M_{r,\text{crit}} \), the resistances stay constant and the particles start to slide, spin or roll against each other. These critical values in the presence of adhesion are given in the following equations
\[ F_{s,\text{crit}} = \mu_f \left| F_{\text{nc}} + 2F_C \right|, \]
\[ M_{r,\text{crit}} = 3\pi a F_{s,\text{crit}}/16, \]
\[ M_{r,\text{crit}} = -4F_C \left( a/a_0 \right)^{3/2} \theta_{\text{crit}} R, \]

where \( \mu_f \) is the friction coefficient and \( \theta_{\text{crit}} \) is the critical angle for the relative rolling of two particles. The model parameters used in Eqs. (27) and (29) can be found in previous literature [36,38] [Chen15, Yang13].

3. Model validation

3.1 3D Duct flow

We firstly test our model with the single phase 3D duct flow. The flow channel is bounded by two pairs of parallel walls at the top, the bottom, the front and the back. Periodic boundary conditions are applied in the other two directions (inlet and outlet). The size of the channel is \( 10 \times 51 \times 51 \) and the fluid density, kinematic viscosity and relaxation parameter are 1,000 kg/m\(^3\), \( 1 \times 10^{-4} \) m\(^2\)/s and 0.65, respectively. The flow is driven by a constant body force, which acts as the pressure gradient. By varying the body force between \( 3.2 \times 10^{-6} \) and \( 6.4 \times 10^{-5} \) (in lattice unit) different channel Reynolds numbers can be achieved. The velocity profile in a duct flow is analytically given as [39] [Drazin06]

\[ U(y,z) = \frac{G}{2\mu} y(l_h - y) - \frac{4Gl_h^2}{\mu\pi^3} \sum_{m=1}^{\infty} \frac{1}{(2m-1)^3} \frac{\sinh(\beta_m y) + \sinh(\beta_m (l_w - z))}{\sinh(\beta_m l_w)} \sin(\beta_m y), \]

where \( \beta_m = \frac{2m-1}{l_h} \pi \), \( l_h \) and \( l_w \) stand for the height and width of the channel, respectively, and
$G$ and $\mu$ represent the pressure gradient and fluid dynamic viscosity, respectively.

Fig. 7 Normalized velocity profiles for duct flow with different Reynolds numbers at the position $x=5$, $y=26$. The inset shows the flow field in the $yz$-plane, where the black solid line indicates the location of the velocity profile shown in the main plot.

Figure 7 shows the velocity profiles for different channel Reynolds numbers, which are normalized with the corresponding maximum velocity along the centerline. It can be seen that the normalized velocity profiles for different channel Reynolds numbers all collapse onto a single curve, which agree perfectly with the theoretical prediction, within a maximum relative error less than 0.5%. Therefore, it indicates that our numerical approach is capable of accurately modelling the single phase fluid flow.
3.2 Drag force on a fixed sphere

It was reported that the immersed moving boundary method could improve the boundary representation and smooth the hydrodynamic forces calculated at an obstacle’s boundary nodes [21] [Owen11]. To test the accuracy in the force calculation, we simulated the flow past a fixed sphere and calculated the drag force. The sphere is fixed in the center of a cuboid box with size of $40d \times 5d \times 5d$, where $d$ is the diameter of the sphere. The box length is longer than 30 radii in order to eliminate the periodic effect in the flow direction [21] [Owen11]. The flow is driven by a constant body force and periodic boundary condition is set along the flow direction. No-slip wall boundary conditions are set in the other four faces of the box. The drag coefficient is calculated as

$$C_d = \frac{8F_d}{\rho U_{\text{max}}^2 \pi d^2},$$

(32)

where $U_{\text{max}}$ is the maximum velocity along the centerline of the box. To check the robustness of the IMB method, we also perform the simulation with the same setup and parameters but use the modified bounce-back (MBB) scheme proposed by Ladd [11] [Ladd94a], where the particle boundary is described with stepwise lattice representation. Here we use three different diameters $d=10, 16, 20$ in the simulation with MBB, aiming at investigating the effect of sphere-to-lattice size resolution, while only one diameter $d=10$ is considered with the IMB method.
Figure 8 shows the drag coefficient $C_d$ as a function of the particle Reynolds number $Re_p$. The solid line refers to a widely accepted empirical law of the drag coefficient proposed by Schiller and Naumann [39] [Schiller35],

$$C_d = \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687}) ,$$  \hspace{1cm} (33)

which remains valid to within 5% of the experimental data for particle Reynolds number up to 800. We can see that the MBB results seem to deviate from the predication. The increase in the
size resolution does not improve the results. However, we can see that the IMB results agree well with the empirical equation, except for some small deviations for low $Re_p$. As a consequence, we conclude that the IMB method is more accurate for the force calculation, which even performs better with a relatively low size resolution.

### 3.3 Flow through porous media

To further validate the IMB method, we perform the simulation of fluid flowing through a porous media consisting of a cubic packing of uniform spheres. The packing is made of 512 particles, forming a sphere array of $8\times8\times8$ and is initially placed in the center of a cuboid box with size of $100\times49\times49$. The particle diameter is $4.8\times10^{-5}$ m, which equals to 6 in lattice unit. The fluid density, kinematic viscosity and relaxation parameter are 1,000 kg/m$^3$, $1\times10^{-6}$ m$^2$/s and 2, respectively. It should be noted that the particles are fixed all the time, which means that their velocities and positions are not updated. The flow is driven by a constant body force, and periodic boundary conditions are set in all the faces of the box. The same simulation with the MBB is also performed to make a comparison. Figure 9 shows the schematic of the simulation setup, where slices of the flow field in $x$-plane and $y$-plane are extracted and compared between the MBB and the IMB methods. The positions of the slices range between $x=41.5$ and $x=47.5$ in the $x$-plane, and between $y=25$ and $y=31$ in the $y$-plane, which exactly covers a whole layer of sphere array.
Fig. 9 Schematic of the slices of the flow field.

Figures 10 and 11 show typical flow field obtained from both IMB and MBB in $x$-plane and $y$-plane, respectively. Intuitively, for the IMB method, the slices of the flow field in $x$-plane gradually changes as they move from one layer to the next (see Figs. 10a and 11a). The channel that allows for the fluid flowing through narrows first and then widens after the slice moves past the center of the sphere. Furthermore, the shape of the channel formed by the four neighbouring spheres is like a star, which looks physically reasonable. More details can be visualized in the slices in $y$-plane. However, when we turn to the MBB results, we find that the slices of the flow field in $x$-plane and $y$-plane are almost identical at different positions (see Figs. 10b and 11b), except for the ones at $y=25, 31$, which correspond to the gap between two layers of sphere arrays. The shape of the channel looks like a cross in $x$-plane and does not
change as the slice moves, which is physically incorrect. Moreover, the spherical feature of particles in the porous media seems to be improperly reflected, since the particles all look like cubes. Due to the stepwise lattice nature, the MBB results can be further improved with the increase of the size resolution. Nevertheless, it will result in a huge increase of the computational cost, because the total amount of calculation is proportional to the third power of the resolution. Therefore, it is demonstrated that the IMB method is a more efficient and economical method.

Fig. 10 (color online, single column) Slices of flow field in $x$-plane obtained from (a) IMB
and (b) MBB, respectively. The color represents the magnitude of the x-velocity, which refers to the same legend in Fig. 9.

![Fig. 11 Slices of flow field in y-plane obtained from (a) IMB and (b) MBB, respectively. The color represents the magnitude of the x-velocity, which refers to the same legend in Fig. 9.](image)

### 3.4 Random packing of microspheres in quiescent fluid

The adhesive contact model is validated through the simulation of random packing of microspheres in a quiescent fluid domain. Initially, 500 uniform spheres are randomly placed in a cuboid box with size of 50×50×100, as shown in Figs. 12a and 12e. Periodic boundary conditions are applied in both x and y directions, while no-slip wall boundaries are set on both top and bottom of the domain. The spheres settle under gravity with an initial velocity of (0.0, 0.0, -0.1) to accelerate the settling, which is in the same direction of the gravity. The diameter of the sphere is 4.8×10⁻⁵ m, equivalent to 6 in lattice unit, and the mass density of the sphere is
3,000 kg/m³. The surface energy of the sphere is fixed at 15 mJ/m² to account for the van der Waals adhesion. The fluid is quiescent at the beginning, with the density and kinematic viscosity of 1,000 kg/m³ and $1 \times 10^{-6}$ m²/s, respectively. The improved IMB method is employed in the simulation with the dimensionless relaxation parameter of 2. After a sufficient long time, all the kinetic energy of the spheres is dissipated and a mechanically stable packing is formed. A comparative simulation with the same configuration but without adhesion is also performed to further validate the adhesion model. Note that the non-adhesive Hertz contact model is used in the comparative simulation.

Figure 12 shows the snapshots of the packing process at different time points for both adhesive and non-adhesive spheres. Generally, the packing formation process looks reasonable for both cases. The packing structure becomes stable after $t=20,000$. More importantly, we can see that the packing structure of the adhesive spheres is higher than that of the non-adhesive spheres, leading to a lower packing fraction, which agrees with the previous findings on random adhesive packings [25,41-42] [Liu15, Liu16, Liu17]. Then we quantify the representative properties, i.e. the global packing fraction and the mean coordination number to make a further comparison. The global packing fraction is defined as the ratio of the total volume of the spheres to the total volume that the packing occupies. The coordination number is defined as the number of the neighbours that are in contact with a reference sphere, which is calculated by judging whether the center-to-center distance of two spheres is smaller than the sum of their radii. The mean coordination number is then obtained through the average over all the spheres.
Fig. 12 (color online, single column) The random packing of 500 spheres in quiescent fluid at (a)(e) $t=0$, (b)(f) $t=5,000$, (c)(g) $t=20,000$, (d)(h) $t=100,000$. (a) (b) (c) (d) denote the adhesive spheres, while (e) (f) (g) (h) stand for the non-adhesive spheres. The contour plot shows the velocity magnitude of the flow field in the slice of $x=25$ in the $x$-plane.

Figure 13a presents the time evolution of the global packing fraction $\phi$ and the mean coordination number $Z$ for both adhesive and non-adhesive spheres. We can see that both $\phi$ and $Z$ rise quickly from $t=0$ to $t=20,000$, during which the particles are still in settling and start to form contact network. After $t=20,000$, the packing structure enters the relaxation stage and gradually becomes stable. However, obvious distinctions can be observed between adhesive and non-adhesive spheres. The global packing fraction and the mean coordination number of adhesive spheres are $\phi = 0.445$ and $Z=4.94$, which are lower than those of non-adhesive
spheres, $\phi = 0.548$ and $Z = 5.52$. The result for non-adhesive sphere is in quantitative agreement with the random loose packing (RLP) limit of granular matter [43] [Onoda90], while the result for adhesive sphere is below the RLP limit, agreeing with the previous investigations [25,44] [Liu15, Martin08]. Further comparison can be found in Fig. 13b, showing the probability distribution function $P(z)$ of the local coordination number of each sphere. For non-adhesive spheres, most of the spheres have 6 neighbours, while for adhesive spheres, the majority of the spheres tends to have only 4 or 5 contacts. This is because adhesion can provide additional resistances to prevent the sphere from rolling over other spheres, thus fewer contacts are needed to realise a local mechanical equilibrium [41] [Liu16]. These observations confirm that our adhesive contact model implemented in the hybrid IMB-LBM-DEM modelling framework is capable of capturing the adhesive mechanics between microspheres effectively.

Fig. 13 (single column) (a) The global packing fraction and mean coordination number as a function of the computational time for both adhesive and non-adhesive spheres. (b) The probability distribution function of the local coordination number.
4. Conclusions

In this paper, an efficient LBM-DEM numerical framework for modelling adhesive particle-liquid flow is developed. An improved IMB method, which properly considers the multi-intersection of more than one particle with the same lattice cell, is implemented for solving the fluid-particle interactions. A linear approximation method is applied in the estimation of the solid fraction in a local lattice cell, which is validated to be of high efficiency and accuracy compared with other approaches. In the DEM, the JKR adhesive contact model along with damping is adopted to account for the particle-particle normal force. Other dissipative interactions, including the sliding, twisting and rolling resistances, are all well considered with a spring-dashpot-slider model in the presence of adhesion. Validated with several benchmark cases, the hybrid IMB-LBM-DEM numerical approach is proved to be capable of providing more detailed and smooth results in the computation of fluid flow between dense particle arrays, as well as capturing the adhesion physics between microspheres. However, considering both the small size scale and short collision time scale of micro-sized particles, this numerical framework might not be suitable for simulations across large length scale. Further improvements in computing efficiency can be realised with parallel algorithms on GPUs or clusters.

Acknowledgement

This work is funded by the Engineering and Physical Sciences Research Council, UK (EPSRC,
Grants No: EP/N033876/1). W. Liu is grateful to Dr. Duo Zhang and Dr. Nicolin Govender in University of Surrey, Prof. Shuiqing Li and Mr. Sheng Chen in Tsinghua University for their helpful suggestions and fruitful discussions in developing the LBM-DEM numerical approach.

References

1. Hu H H, Joseph D D, Crochet M J. Direct simulation of fluid particle motions. Theoretical and Computational Fluid Dynamics, 1992, 3(5): 285-306. https://doi.org/10.1007/BF00717645

2. Feng J, Hu H H, Joseph D D. Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid Part 1. Sedimentation. Journal of Fluid Mechanics, 1994, 261: 95-134. https://doi.org/10.1017/S0022112094000285

3. Feng J, Hu H H, Joseph D D. Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 2. Couette and Poiseuille flows. Journal of Fluid Mechanics, 1994, 277: 271-301. https://doi.org/10.1017/S0022112094002764

4. Kuipers J A M, Van Duin K J, Van Beckum F P H, et al. A numerical model of gas-fluidized beds. Chemical Engineering Science, 1992, 47(8): 1913-1924. https://doi.org/10.1016/0009-2509(92)80309-Z

5. Tsuji Y, Kawaguchi T, Tanaka T. Discrete particle simulation of two-dimensional fluidized bed. Powder Technology, 1993, 77(1): 79-87. https://doi.org/10.1016/0032-5910(93)85010-7

6. Kafui K D, Thornton C, Adams M J. Discrete particle-continuum fluid modelling of gas-
solid fluidised beds. Chemical Engineering Science, 2002, 57(13): 2395-2410. https://doi.org/10.1016/S0009-2509(02)00140-9

7. Zhu H P, Zhou Z Y, Yang R Y, et al. Discrete particle simulation of particulate systems: theoretical developments. Chemical Engineering Science, 2007, 62(13): 3378-3396. https://doi.org/10.1016/j.ces.2006.12.089

8. Chen H, Chen S, Matthaeus W H. Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. Physical Review A, 1992, 45(8): R5339. https://doi.org/10.1103/PhysRevA.45.R5339

9. Chen S, Doolen G D. Lattice Boltzmann method for fluid flows. Annual Review of Fluid Mechanics, 1998, 30(1): 329-364. https://doi.org/10.1146/annurev.fluid.30.1.329

10. Wolf-Gladrow D A. Lattice-gas cellular automata and lattice Boltzmann models: an introduction. Springer, 2004.

11. Ladd A J C. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. Journal of Fluid Mechanics, 1994, 271: 285-309. https://doi.org/10.1017/S0022112094001771

12. Ladd A J C. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results. Journal of Fluid Mechanics, 1994, 271: 311-339. https://doi.org/10.1017/S0022112094001783

13. Peskin C S. Numerical analysis of blood flow in the heart. Journal of Computational Physics, 1977, 25(3): 220-252. https://doi.org/10.1016/0021-9991(77)90100-0

14. Peskin C S. The immersed boundary method. Acta Numerica, 2002, 11: 479-517. https://doi.org/10.1017/S0962492902000077
15. Feng Z G, Michaelides E E. The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. Journal of Computational Physics, 2004, 195(2): 602-628. https://doi.org/10.1016/j.jcp.2003.10.013

16. Noble D R, Torczynski J R. A lattice-Boltzmann method for partially saturated computational cells. International Journal of Modern Physics C, 1998, 9(08): 1189-1201. https://doi.org/10.1142/S0129183198001084

17. Feng Y T, Han K, Owen D R J. Coupled lattice Boltzmann method and discrete element modelling of particle transport in turbulent fluid flows: Computational issues. International Journal for Numerical Methods in Engineering, 2007, 72(9): 1111-1134. https://doi.org/10.1002/nme.2114

18. Mittal R, Iaccarino G. Immersed boundary methods. Annual Review of Fluid Mechanics, 2005, 37: 239-261. https://doi.org/10.1146/annurev.fluid.37.061903.175743

19. Strack O E, Cook B K. Three-dimensional immersed boundary conditions for moving solids in the lattice-Boltzmann method. International Journal for Numerical Methods in Fluids, 2007, 55(2): 103-125. https://doi.org/10.1002/fld.1437

20. Feng Y T, Han K, Owen D R J. Combined three-dimensional lattice Boltzmann method and discrete element method for modelling fluid-particle interactions with experimental assessment. International Journal for Numerical Methods in Engineering, 2010, 81(2): 229-245. https://doi.org/10.1002/nme.2689

21. Owen D R J, Leonardi C R, Feng Y T. An efficient framework for fluid–structure interaction using the lattice Boltzmann method and immersed moving boundaries. International Journal for Numerical Methods in Engineering, 2011, 87(1-5): 66-95.
22. Han Y, Cundall P A. LBM-DEM modeling of fluid–solid interaction in porous media. International Journal for Numerical and Analytical Methods in Geomechanics, 2013, 37(10): 1391-1407. https://doi.org/10.1002/nme.2985

23. Li S, Marshall J S, Liu G, et al. Adhesive particulate flow: The discrete-element method and its application in energy and environmental engineering. Progress in Energy and Combustion Science, 2011, 37(6): 633-668. https://doi.org/10.1016/j.pecs.2011.02.001

24. Marshall J S, Li S. Adhesive particle flow. Cambridge University Press, 2014.

25. Liu W, Li S, Baule A, et al. Adhesive loose packings of small dry particles. Soft Matter, 2015, 11(32): 6492-6498. https://doi.org/10.1039/C5SM01169H

26. Chen S, Li S, Liu W, et al. Effect of long-range repulsive Coulomb interactions on packing structure of adhesive particles. Soft Matter, 2016, 12(6): 1836-1846. https://doi.org/10.1039/C5SM02403J

27. Chen S, Liu W, Li S. Effect of long-range electrostatic repulsion on pore clogging during microfiltration. Physical Review E, 2016, 94(6): 063108. https://doi.org/10.1103/PhysRevE.94.063108

28. Dominik C, Tielens A. The physics of dust coagulation and the structure of dust aggregates in space. The Astrophysical Journal, 1997, 480(2): 647. https://doi.org/10.1086/303996

29. Kinch K M, Sohl-Dickstein J, Bell J F, et al. Dust deposition on the Mars Exploration Rover Panoramic Camera (Pancam) calibration targets. Journal of Geophysical Research: Planets, 2007, 112(E6). https://doi.org/10.1029/2006JE002807

30. Bhatnagar P L, Gross E P, Krook M. A model for collision processes in gases. I. Small
amplitude processes in charged and neutral one-component systems. Physical Review, 1954, 94(3): 511. https://doi.org/10.1103/PhysRev.94.511

31. Qian Y H, d'Humières D, Lallemand P. Lattice BGK models for Navier-Stokes equation. Europhysics Letters, 1992, 17(6): 479. https://doi.org/10.1209/0295-5075/17/6/001

32. Guo Z, Zheng C, Shi B. Discrete lattice effects on the forcing term in the lattice Boltzmann method. Physical Review E, 2002, 65(4): 046308. https://doi.org/10.1103/PhysRevE.65.046308

33. He X, Zou Q, Luo L S, et al. Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model. Journal of Statistical Physics, 1997, 87(1-2): 115-136. https://doi.org/10.1007/BF02181482

34. Jones B D, Williams J R. Fast computation of accurate sphere-cube intersection volume. Engineering Computations, 2017, 34(4): 1204-1216. https://doi.org/10.1108/EC-02-2016-0052

35. Liu G, Li S, Yao Q. A JKR-based dynamic model for the impact of micro-particle with a flat surface. Powder Technology, 2011, 207(1-3): 215-223. https://doi.org/10.1016/j.powtec.2010.11.002

36. Chen S, Li S, Yang M. Sticking/rebound criterion for collisions of small adhesive particles: Effects of impact parameter and particle size. Powder Technology, 2015, 274: 431-440. https://doi.org/10.1016/j.powtec.2015.01.051

37. Johnson K L, Kendall K, Roberts A D. Surface energy and the contact of elastic solids. Proceedings of the Royal Society of London. A. mathematical and physical sciences, 1971, 324(1558): 301-313. https://doi.org/10.1098/rspa.1971.0141
38. Yang M, Li S, Yao Q. Mechanistic studies of initial deposition of fine adhesive particles on a fiber using discrete-element methods. Powder Technology, 2013, 248: 44-53. https://doi.org/10.1016/j.powtec.2012.12.016

39. Drazin P G, Riley N. The Navier-Stokes equations: a classification of flows and exact solutions. Cambridge University Press, 2006.

40. Schiller L, Naumann A. A drag coefficient correlation, Zeitschrift des Vereins Deutscher Ingenieure, 1935, 77: 318-320.

41. Liu W, Li S, Chen S. Computer simulation of random loose packings of micro-particles in presence of adhesion and friction. Powder Technology, 2016, 302: 414-422. https://doi.org/10.1016/j.powtec.2016.08.068

42. Liu W, Jin Y, Chen S, et al. Equation of state for random sphere packings with arbitrary adhesion and friction. Soft Matter, 2017, 13(2): 421-427. https://doi.org/10.1039/C6SM02216B

43. Onoda G Y, Liniger E G. Random loose packings of uniform spheres and the dilatancy onset. Physical Review Letters, 1990, 64(22): 2727. https://doi.org/10.1103/PhysRevLett.64.2727

44. Martin C L, Bordia R K. Influence of adhesion and friction on the geometry of packings of spherical particles. Physical Review E, 2008, 77(3): 031307. https://doi.org/10.1103/PhysRevE.77.031307