Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 195 (2004) 602–628
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems

Zhi-Gang Feng, Efstathios E. Michaelides *

School of Engineering and SCRC of National Institute for Global Environmental Change, Tulane University, New Orleans, LA 70118, USA

Received 19 February 2003; received in revised form 25 August 2003; accepted 10 October 2003

Abstract

A new computational method, the immersed boundary-lattice Boltzmann method, is presented. This method is a combination and utilizes the most desirable features of the lattice Boltzmann and the immersed boundary methods. The method uses a regular Eulerian grid for the flow domain and a Lagrangian grid to follow particles that are contained in the flow field. The rigid body conditions for the fluid and the particles are enforced by a penalty method, which assumes that the particle boundary is deformable with a high stiffness constant. The velocity field of the fluid and particles is solved by adding a force density term into the lattice Boltzmann equation. This novel method preserves the advantages of LBM in tracking a group of particles and, at the same time, provides an alternative and better approach to treating the solid–fluid boundary conditions. The method also solves the problems of fluctuation of the forces and velocities on the particles when the ''bounce-back'' boundary conditions are applied. This method enables one to simulate problems with particle deformation and fluid–structure deformation. Its results are validated by comparison with results from other methods.

1. Introduction

The particle–fluid interaction problem has widely varying applications in the fields of chemical, aerospace and environmental engineering as well as in geology and biology. Applications ranging from the transport of radionuclides by sedimentary particles in aquatic environments, to fluidized bed reactors, to droplet formation and combustion depend on our knowledge of the parameters that govern the interactions of particles and fluids. Because of the importance of these applications, the fluid–particles interaction problems have been attracting considerable attention, both experimentally and numerically.

Conventional numerical methods, such as the finite volume method (FVM) and finite element method (FEM), have limited success in the simulations of particulate flows with a high number of particles,

* Corresponding author. Tel.: +1-504-865-5764/5819; fax: +1-504-862-8747.
E-mail address: emichael@tulane.edu (E.E. Michaelides).

0021-9991/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2003.10.013

especially in three-dimensional simulations. The main obstacle with these methods is the need to generate new, geometrically adapted grids, which is a very time-consuming task especially in three-dimensional flows. Methods such as Stokesian dynamics (SD) that was developed by Brady and Bossis [1] and the boundary element method (BEM) have been used successfully to simulate particulate flows at creeping flow conditions. However, both of these methods are very complicated to code and are only valid at low Reynolds numbers. Moreover, it is extremely difficult to extend the SD method to non-spherical particles or flow systems with a boundary. More recently, a new method, PHYSALIS, has been proposed by Zhang and Prosperetti [21]. This method incorporates the analytical solutions for the region near the particle surface, with some parameters determined by matching the outer flow conditions. The method requires the existence of analytical solutions for the particles, which are normally given for particles with simple shapes and very low Reynolds numbers.

In the early 1990, Ladd [13,14] successfully applied the lattice Boltzmann method (LBM) to particle–fluid suspensions. Since then the LBM has proven to be a robust and efficient method to simulate particulate flows with a large number of particles [5,6,15]. The LBM overcame the limitations of the conventional Finite Volume and Finite Element Methods by using a fixed, non-adaptive (Eulerian) grid system to represent the flow field. Because the LBM does not require re-meshing, it is easier to code and has proven to be more efficient computationally.

When the LBM is used to simulate particle–fluid interaction problems, the no-slip condition on the particle–fluid interface is treated by the bounce-back rule [14] and the particle surface is represented by the so-called boundary nodes, which are essentially a set of the mid-points of the links between two fixed grids. One of the boundary nodes is within the fluid domain and the other is within the solid domain. This arrangement makes it necessary to use a large number of lattice grids for the particles if one is to represent accurately their physical boundaries. Also, the finite number of boundary nodes makes necessary the stepwise representation of the particle boundary. This causes fluctuations on the computation of forces on the particle and limits the ability of LBM to solve particle–fluid interaction problems at high Reynolds numbers. When a particle moves, its computational boundary will vary and this also causes fluctuations in the resulting computation of forces and velocities of the particle.

In the 1970s, Peskin [19] developed a novel numerical method called the immersed boundary method in order to model the flow of blood in the heart. Such a flow is regulated by heart valves, which are moving boundaries immersed in the fluid (blood). This method uses a fixed Cartesian mesh for the fluid. However, for the boundaries that are immersed in the fluid, the method uses a set of boundary points, which may be advected by the fluid interaction. This method is especially suitable for the simulation of the deformation of immersed boundaries by fluid–structure interaction, and it has been widely used in biological fluid dynamics. Fogelson and Peskin [7] have showed that this method could also be employed to simulate flows with suspended particles. Höfler and Schwarzer [12] presented a finite-difference method for particle-laden flows by adding a constraint force into the Navier–Stokes equations to enforce particle rigid motions.

The basic idea of the immersed boundary method (IBM) as applied to particulate flows is to treat the particle boundary as deformable, but with high stiffness. A small distortion of the particle boundary will yield a force that tends to restore the particle into its original shape. The balances of such forces, together with the other external forces exerted on the particle, are distributed into the Eulerian nodes of the grid and the Navier–Stokes equations with a body force are solved over the whole fluid–particles domain.

The concept of IBM has been employed into the FEM. Glowinski and his colleagues [10,11,17,18] have developed the fictitious domain method (FDM) by using Lagrange multipliers to enforce the no-slip boundary conditions between particle surfaces and fluid. They were able to apply this method in order to simulate a flow system with 1024 spherical particles [11].

In this study we have adopted the same approach of the IBM to simulate the rigid particle motion by using the LBM in order to solve for the fluid flow. The key point of the success of both LBM and IBM is that instead of re-meshing the fluid domain, both methods use a fixed mesh to represent the fluid field. In

the LBM, the moving boundaries are approximated by fixed points on the grid (these are actually the midpoints of the boundary links if the bounce-back rule is used to implement the no-slip boundary condition), that is, the moving boundaries are described by Eulerian points. In the IBM, the moving boundaries are represented by a set of boundary nodes, which are moving with the fluid, that is, the moving boundaries are described by Lagrangian points. This combined method may be called immersed boundary-lattice Boltzmann method (IB-LBM). We use the lattice grid for the fluid flow field, and use the boundary points to represent particle surfaces. The rigid body condition is approximated by considering the particle boundary to be a shell with high stiffness. Any small deformation on this shell results in a large force that restores the particle boundary into its original shape. The deformation is calculated by comparing the boundary point (tracer) and the reference point (mark) that undergoes rigid body motions with particles. Hence, a lattice Boltzmann equation with body forces is solved to obtain the fluid velocity.

In the first part of our paper, we give a general description of the proposed novel method, IB-LBM, and the collision rule we use. To validate the method, we present the simulation of the migration of a neutrally buoyant particle in a simple shear flow and compare the results with results obtained by others using the FEM and the LBM. Subsequently, we solve the problem of the flow of two interacting circular particles settling in a channel, the so-called drafting–kissing–tumbling (DKT) problem, and provide an extensive analysis of the parameters used in the IB-LBM. Finally, we apply this method to simulate the sedimentation of a large number of circular particles in an enclosure.

2. Numerical method

2.1. A description of the IB-LBM

The LBM has been developed from the lattice-gas automata [8,9], and its application to the motion of solid particles suspended in a fluid was first introduced by Ladd [13,14]. Since then, the LBM has been employed by many researchers and it has been proven to be a robust method for the solution of particle–fluid interaction problems [5,6,15].

The LBM uses a regular grid and decomposes the fluid domain into a set of lattice nodes. The fluid is modeled as a group of fluid particles that are only allowed to move between lattice nodes or stay at rest. The composition of the lattice nodes depends on the chosen lattice model. The most common lattice model for two-dimensional simulations is the one using a square lattice with nine discrete velocity directions, while the model for three-dimensional simulations uses a cubic lattice with fifteen discrete velocity directions [5].

The motion of fluid particles is described by the discrete lattice-Boltzmann equation, which is

$$n_{\sigma i}(\mathbf{x}+\mathbf{e}_{\sigma i}, t+1) - n_{\sigma i}(\mathbf{x}, t) = -\frac{1}{\tau}\left[n_{\sigma i}(\mathbf{x}, t) - n^{(0)}_{\sigma i}(\mathbf{x}, t)\right], \tag{1}$$

where $n^{(0)}_{\sigma i}(\mathbf{x}, t)$ is the equilibrium distribution function, τ is the relaxation time, and t is the lattice simulation time. The notation for the subscripts are as follows: $\sigma = 1$ corresponds to the fluid particles moving to their four nearest neighbors along the axial directions; $i = 1$–4 represents these neighboring particles. The value $\sigma = 2$ signifies that the fluid particles move to their second nearest neighbors along the diagonal directions, with $i = 5$, 6, 7 and 8 representing these neighbors. Finally, $\sigma = 0$ and $i = 0$ correspond to fluid particles being at rest.

The equilibrium distribution function, $n^{0}_{\sigma i}(\mathbf{x}, t)$, which is determined by the local velocity at a lattice node only, may be written as follows:

$$n^{(0)}_{\sigma i}(\mathbf{x}, t) = \rho w_i\left[1 + 3(\mathbf{e}_{\sigma i}\cdot\mathbf{u}) + \tfrac{9}{2}(e_{\sigma i}\cdot\mathbf{u})^2 + \tfrac{3}{2}(\mathbf{u}\cdot\mathbf{u})\right], \tag{2}$$

where the values of the weights are: $w_0 = 4/9,\ w_1 = w_3 = w_5 = w_7 = 1/9$, and $w_2 = w_4 = w_6 = w_8 = 1/36$. The vectorial function $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the velocity at the lattice node and $\mathbf{e}_{\sigma i}$ is the vector that represents the nine directions mentioned above.

The relaxation time, τ, is related to the kinematic viscosity of the fluid by the following expression:

$$v = (2\tau - 1)/6. \tag{3}$$

The values for the relaxation time chosen in the simulations render the viscosity of the fluid equal to the viscosity of pure water. The choice of the viscosity also affects the time step of the computations through the following relationship:

$$\Delta t = (\Delta x)^2 \frac{v}{v_r}, \tag{4}$$

where Δx is the physical grid size and v_r is the physical fluid kinematic viscosity.

The application of Eqs. (1) and (2) in incompressible flows results in a computational error, which is proportional to the square of a Mach number, Ma^2, defined as follows [13]:

$$Ma \equiv \frac{U_c}{\Delta x/\sqrt{3}\Delta t} = \frac{U_c \Delta x}{v_r}\left(\frac{2\tau - 1}{2\sqrt{3}}\right), \tag{5}$$

where U_c is the characteristic velocity of the actual flow. Hence, one of the criteria for the choice of the relaxation time to be is ensure that $Ma \ll 1$. The numerical choice of τ in the simulations always satisfies $Ma \leqslant 0.1$.

At each node of the lattice, the density and momentum can be written in terms of the distribution function, $n_{\sigma i}(\mathbf{x}, t)$:

$$\rho(\mathbf{x}, t) = \sum_{\sigma, i} n_{\sigma i}(x, t) \tag{6}$$

and

$$\rho(\mathbf{x}, t)\mathbf{u} = \sum_{\sigma, i} n_{\sigma i}(\mathbf{x}, t)\mathbf{e}_{\sigma i}. \tag{7}$$

In order to numerically simulate the hydrodynamic interactions between the solid particles and the fluid, the LBM must be modified to incorporate the boundary conditions imposed on the fluid by the solid particles. This represents the greatest challenge of using the method in the solution of fluid–particle interaction problems. The technique available to deal with the solid–fluid boundary condition is to apply the so-called bounce-back rule [13,14], or one of the modified versions that emanate from this rule [15].

Given that the particles are spherical or circular and the lattice nodes rectangular, the boundary surfaces of the particles intercept several lattice links that connect the nodes of the flow field between those within the fluid and those within the particles. These boundary nodes are defined as the midpoints along the intercepted links and their set approximates the particle surface. Obviously, in order to have a more accurate representation of the shape of a particle, the dimension of the particles must be considerably larger than the dimension of the lattice and this necessitates a large number of the lattice grid points. Under these conditions, the no-slip boundary condition on the surface of the moving particles is satisfied by the requirement that the fluid velocity has the same value at the boundary nodes as the particle velocity. It must be emphasized that the definition of the boundary nodes in the LBM and the IBM is different: in the IBM the boundary node is a Lagrangian node, which is attached to a moving fiber or particle.

In order to account for the momentum exchange when the particle relative velocity is not zero, a fluid particles collision function is used, which is given by the following equation:

$$n_{\sigma i'}(\mathbf{x}, t+1) = n_{\sigma i}(\mathbf{x}, t_+) - 2w_2(\mathbf{e}_{\sigma i} \cdot \mathbf{U}_b), \tag{8}$$

where $\mathbf{x}$ is the position of the node adjacent to the solid-surface, $\mathbf{U}_b$ is the velocity of the boundary node, $\sigma i'$ denotes the direction opposite to the incident direction σi (oftentimes called the "reflection direction"), and t^+ is the post-collision time. Hence, the hydrodynamic force exerted on the solid particle at the boundary node is given by the following expression:

$$\mathbf{F}\left(\mathbf{x} + \tfrac{1}{2}\mathbf{e}_{\sigma i}, t\right) = 2\mathbf{e}_{\sigma i}[f_{\sigma i}(\mathbf{x}, t_+) - 2w_2(\mathbf{U}_b \cdot \mathbf{e}_{\sigma i})]. \tag{9}$$

The total force $\mathbf{F}_t$ and torque $\mathbf{T}_t$ on the solid particles can be obtained by summing all the forces and torques, respectively. Once the forces on the particles are calculated, both the translational and rotational velocities of the particles are updated explicitly. However, the solid particle moves in a continuous way, based on Newtonian dynamics. The position of the solid particle is then updated.

In the application of the LBM the distortion between the simulation boundary and the physical boundary of a particle depends on the particle size when it is given in terms of lattice units. If the diameter of the particle is composed of a large number of lattice units, the boundary is smooth and the computation of the hydrodynamic force is accurate. On the contrary, in the case where the particle diameter corresponds to only a few lattice units, the particle outline is rough and an effective hydrodynamic radius, instead of the physical radius, is needed for more accurate computations. The effective radius can be determined numerically from simulations of the flow around an isolated particle and subsequent comparison of the drag force between the numerical results and a known analytical solution [14]. The drawback of this method is that most analytical solutions are only valid at very low Reynolds numbers and that there are cases, such as particles moving in a pressure-driven flow, where no accurate analytical solutions are available [16]. In addition, the treatment of the solid–fluid boundary by the bounce-back rule causes fluctuations on the calculated forces on the particles and sometimes fails to work at high Reynolds numbers because of the distortion that occurs near the particle boundary.

In order to demonstrate the above, we consider a particle of diameter 7 lattice units at two different times t_1 and t_2, as illustrated in Fig. 1. The dark solid line is the physical boundary of the particle, and the thick curves composed by the boundary nodes are the computational boundary based on bounce-back rules of the LBM. It is obvious that the two computational boundaries differ at time t_1 and t_2, and this has an effect on the computed forces and velocities. This is an inherent drawback of the LBM when applied to particle–fluid interaction problems and, for this reason, one has to use a finer grid to minimize such fluctuations.

A drawback with the conventional LBM when it is applied to a large number of particles is that it uses regular nodes to represent the particle surface, as the particles move inside the fluid. One may resolve this problem by using a set of independent nodes that are attached to the boundary in order to represent the particle surface. That is, instead of using the Eulerian nodes that are set for the fluid domain, the particle

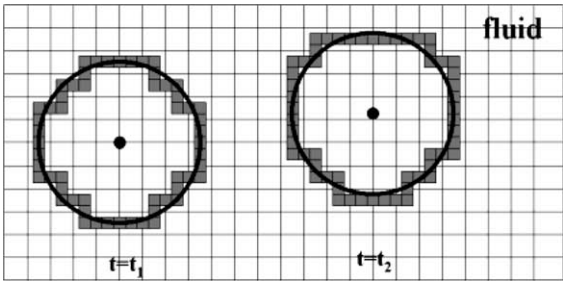

Fig. 1. Particle computational boundaries at two times, t_1 and t_2.

boundary may be represented by a set of Lagrangian points that are advected by the fluid. This is the basic concept of the IBM or FDM.

In this paper, we introduce this concept of IBM into the LBM in order to simulate particle and fluid interaction problems. By doing this, we are able to use the advantages of both the LBM and the IBM. This new method, called the immersed boundary-lattice Boltzmann method, or IB-LBM, apart from being an improvement to the regular LBM may also be used to solve problems with particle deformation, which cannot be handled with the traditional LBM.

For the development of the IB-LBM concept, we consider a flexible fiber or particle with a boundary curve, Γ, immersed in a two-dimensional incompressible viscous fluid with a domain, Ω. The fiber boundary, Γ, is represented by the Lagrangian parametric coordinates, $\mathbf{s}$, and the flow domain, Ω, is represented by the Eulerian coordinates $\mathbf{x}$. Hence, any position on the fiber or particle may be written as $\mathbf{x} = \mathbf{X}(\mathbf{s}, t)$. Let $\mathbf{F}(\mathbf{s}, t)$ and $\mathbf{f}(\mathbf{x}, t)$ represent the fiber force density and the fluid body force density. The fiber force on each boundary node is the combination of the internal link forces and the external forces, such as the gravity force, and the inter-particle force. The internal link force is a function of the boundary deformation. For elastic materials, it is given by a generalized form of Hooke's law.

In order to satisfy the no-slip boundary condition on the particle–fluid interface, the velocity on the fiber/particle must be the same as its neighboring fluid velocity, that is, we must have the condition:

$$\frac{\partial \mathbf{X}(\mathbf{s}, t)}{\partial t} = \mathbf{u}(\mathbf{X}(\mathbf{s}, t), t), \tag{10}$$

where $\mathbf{u}$ is the fluid velocity.

Hence, we may write the governing equations for the fluid-fiber composite as follows:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = \mu \nabla^2 \mathbf{u} - \nabla p + \mathbf{f}, \tag{11}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{12}$$

$$\mathbf{f}(\mathbf{x}, t) = \int_{\Gamma} \mathbf{F}(\mathbf{s}, t) \delta(\mathbf{x} - \mathbf{X}(\mathbf{s}, t)) \, \mathrm{d}\mathbf{s}, \tag{13}$$

and

$$\frac{\partial \mathbf{X}}{\partial t} = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(\mathbf{s}, t)) \, \mathrm{d}\mathbf{x}, \tag{14}$$

where $p(\mathbf{x}, t)$ is the fluid pressure, ρ is the fluid density, and μ the fluid viscosity.

Eqs. (11) and (12) are the Navier–Stokes equations of a viscous incompressible flow. Eq. (13) states that the force density of the fluid, $\mathbf{f}(\mathbf{x}, t)$, is obtained from the immersed boundary force density, $\mathbf{F}(\mathbf{s}, t)$, through the integration over the immersed boundary. Eq. (14) is essentially the no-slip condition at the interface, since the fiber/particle moves at the same velocity as the neighboring fluid.

In the numerical scheme of IBM, the whole fluid domain including the parts that are occupied by immersed bodies, such as particles, is divided into a set of fixed lattice nodes. Since these fluid nodes are not moving with the flow, we will call them Eulerian nodes. The immersed boundary is discretized into a group of boundary points that move under the action of the moving fluid. We will call these boundary nodes Lagrangian nodes. It must be pointed out that in the IBM the Lagrangian nodes do not necessarily coincide with the Eulerian nodes.

To employ the IBM concept within the LBM context, the LBM should be able provide a solution for the Navier–Stokes equations with an external force term. Thus, in order to be able to solve the fluid field with body force $\mathbf{f}(\mathbf{x},t)$ included, Eq. (1) should be modified accordingly. There are several ways to include the contribution of an external force into the lattice Boltzmann equation that were reviewed by Buick and Greated [4]. In this paper, the external force is introduced by adding a term to the collision function. This modifies the distribution function and the Boltzmann equation as follows:

$$n_{\sigma i}(\mathbf{x}+\mathbf{e}_{\sigma i},t+1)-n_{\sigma i}(\mathbf{x},t)=-\frac{1}{\tau}\left[n_{\sigma i}(\mathbf{x},t)-n_{\sigma i}^{(0)}(\mathbf{x},t)\right]+\frac{3}{2}w_i\mathbf{f}\cdot\mathbf{e}_{\sigma i}. \tag{15}$$

In order to implement the rigid body motion of a particle in a fluid, Fogelson and Peskin [7] considered the particle boundary as an elastic fiber with high stiffness. Höfler and Schwarzer [12] applied the "marker technique" to simulate the interaction of fluid and rigid particles. We basically employ the same approach, which may be summarized as follows: for each particle, there is a template that is undergoing a rigid body motion consisting of the reference nodes. Before the particle deforms, the reference nodes are coincident with the particle boundary nodes and, hence, the template represents the particle. Once the particle has deformed, because of the particle–fluid interaction, the particle boundary nodes are compared to the reference nodes. Any distortion between the boundary nodes and reference nodes causes an elastic restoration force on the particle boundary node. This approach is shown schematically in Fig. 2. It must be pointed out that this figure is only a schematic diagram and that, in the actual simulations, the distortion is much smaller. Most importantly, the template undergoes a rigid body motion consistent with the particle's translational and rotational velocities. Therefore, the motion of the reference nodes is caused only by the translational and rotational movement of the template, which coincides with the particle.

In order to model this description of the physical surface of the particle, we consider a system with N physical particles. At time t, we assume that the center of the ith particle is at $\mathbf{x}_i(t)$, and the instantaneous particle rotational matrix in $\mathbf{R}_i(t)$. The position of a reference point j in a Lagrangian coordinate system with respect to the particle i may be determined using the following formula:

$$\mathbf{x}_{ij}^{\mathrm{r}}(t)=\mathbf{x}_i(t)+\mathbf{R}_i(t)[\mathbf{x}_{ij}^{\mathrm{r}}(0)-\mathbf{x}_i(0)]. \tag{16}$$

The boundary point $\mathbf{x}_{ij}$ corresponding to the reference point $\mathbf{x}_{ij}^{\mathrm{r}}$ is allowed to be slightly deformed by the fluid. However, when the reference point and the boundary point are not at the same position, that is, when the displacement $\boldsymbol{\xi}_{ij}=\mathbf{x}_{ij}-\mathbf{x}_{ij}^{\mathrm{r}}$ is not zero, a restoration force is developed to restore the boundary point

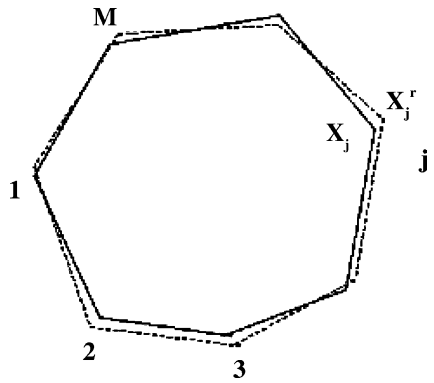

Fig. 2. A particle boundary connected by M boundary points. The dashed lines represent the links between the reference points.

back to the reference point. This restoration force, which is due to the displacement between the reference point and the boundary point, is modeled by a linear spring relation:

$$\mathbf{F}_{ij} = -k\boldsymbol{\xi}_{ij}, \tag{17}$$

where k is the spring constant. We can also write the restoration force as follows:

$$\mathbf{F}_{ij} = \begin{cases} 0, & \|\boldsymbol{\xi}_{ij}\| = 0, \\ -\frac{c_{ij}}{\varepsilon_d}\frac{\boldsymbol{\xi}_{ij}}{\|\boldsymbol{\xi}_{ij}\|}, & \|\boldsymbol{\xi}_{ij}\| > 0. \end{cases} \tag{18}$$

The parameter c_{ij} is the force scale factor and ε_d is the stiffness scale factor for particle deformation. It must be noted that the spring constant is proportional to the inverse of ε_d. The actual value of ε_d used in the simulations should make the particle surface stiff enough for the displacement to be small but not large enough that would affect the convergence of the computations. Our simulations results have found that there is no significant effect to the simulation results by choosing different values for the stiffness scale factor ε_d as long as the particle deformation is small. This type of "penalty method" has also been used by Fogelson and Peskin [7] and Höfler and Schwarzer [12]. Höfler and Schwarzer [12] have included a discussion on their choice of values for the sprint constant k that used in their simulations. It must also be pointed out that if someone wishes to include a damping term with Eq. (17), the computational method will accommodate the term.

For the problems of particle sedimentations considered in this paper as well as for most particulate flow applications, the buoyancy force is a reasonable choice for the scaling force. Therefore:

$$c_{ij} = \rho_f(\sigma - 1)V_P g, \tag{19}$$

where V_P is the volume of the particle and g is the gravitational acceleration. In a more general case of particulate flows, the scaling force is user-defined and should be of the order of magnitude of the characteristic force for the process acting on one particle. Electric and magnetic forces are such examples of forces. In most cases, however, the gravitational force would be a good approximation or choice of this force.

If we assume that the boundary surface is divided into M equal boundary segments and the average deformation ratio of each segment is ε, the total magnitude of the force due to the deformation of the particle surface segments should be comparable to the characteristic force:

$$M\frac{c_{ij}}{\varepsilon_d}\varepsilon \sim C_0 c_{ij}, \tag{20}$$

where C_0 is a dimensionless constant and should in the order of 1. The stiffness scale factor ε_d may be written as follows:

$$\varepsilon_d = \frac{M\varepsilon}{C_0}. \tag{21}$$

The sensitivity of the stiffness scale factor ε_d to the final results will be presented later.

It must be pointed out that the distortion between the physical boundary and the deformed boundary is very small. For example, if we consider a circular particle with radius a and M boundary nodes, the length of each segment is equal to $2\pi a/M$. With a segment deformation ratio ε, the distortion between the deformed segment (the segment connected by the two neighboring reference nodes) and the original segment is equal to $(2\pi\varepsilon/M)a$. In the case $M = 160$ and $\varepsilon = 2.5\%$, the actual distortion of a segment is only 0.1% of the particle radius a. In addition, the distance between one boundary node and its corresponding reference

node is even smaller. Therefore, one may conclude from this that the deformed boundary is very close to the actual physical boundary of the particle.

The force acting on a node, j, due to the ith particle's surface consists of an internal force $\mathbf{F}_{ij}^{\text{int}}$, which is generated by the displacement of the reference point and the boundary point, and of an external forces $\mathbf{F}_{ij}^{\text{ext}}$, which is generated by the physical forces acting on the particle, such as the gravity force and other inter-particle forces. Fogelson and Peskin [7] introduced another method to compute the internal force implicitly in order to avoid possible instabilities. However, in the simulations that will be presented we did not observe any instability and calculated the internal particle forces explicitly by the use of Eq. (18).

According to the marker scheme we employed, the force on each boundary node, which is used to calculate the fluid force density, is the constrained force and may be obtained from Eq. (13). The Dirac delta distribution function for Eq. (13) is used to distribute this force on the boundary point $\mathbf{x}_{ij}$ into the nearby Eulerian grid. In an actual implementation of the method, the force density $\mathbf{f}_{ij}(\mathbf{x})$ is calculated using the following discretized expression:

$$\mathbf{f}_{ij}(\mathbf{x},t)=\sum_{k=1}^{N}\sum_{l=1}^{M}\mathbf{F}_{kl}^{\text{int}}D_{ij}(\mathbf{x}-\mathbf{x}_{kl}(t)), \tag{22}$$

where D_{ij} is a continuous kernel distribution function that approximates the delta function. The choice of this function has to meet certain criteria, which were stipulated by Peskin [20]. For the two-dimensional simulations we choose the following equations:

$$\delta(r)=\begin{cases}\frac{1}{4}\left(1+\cos\left(\frac{\pi|r|}{2}\right)\right), & |r|\leqslant 2,\\ 0, & |r|>2\end{cases} \tag{23}$$

and

$$D_{ij}(\mathbf{x}-\mathbf{x}_{ij})=\delta(x-x_{ij})\delta(y-y_{ij}). \tag{24}$$

When the IB-LBM is used to simulate the particle motions in the fluid, the force and angular momentum should be computed in order to determine the particle motion. As mentioned above, the force on a particle includes both the external forces, such as the gravity/buoyancy force, and the internally developed forces, such as the particle collision forces, $\mathbf{F}_i^{\text{col}}$. The internal link force in the simulation of the rigid particle motion is simply the balance force that was mentioned above. Hence, the total force exerted on the ith particle may be written as follows:

$$\mathbf{F}_i=\left(1-\frac{\rho_{\text{f}}}{\rho_{\text{s}}}\right)M_i\mathbf{g}+\sum_{j=1}^{M}k\boldsymbol{\xi}_{ij}+\mathbf{F}_i^{\text{col}}, \tag{25}$$

in which M_i is the mass of the ith particle.

The torque on the particle is computed by using the following expression:

$$\mathbf{T}_i=\sum_{j=1}^{M}(\mathbf{x}_{ij}-\mathbf{x}_i)\times k\boldsymbol{\xi}_{ij}. \tag{26}$$

Once we have the forces on the particle, the particle motion is determined by solving the following set of equations:

$$M_i\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t}=\left(1-\frac{\rho_{\text{f}}}{\rho_{\text{s}}}\right)M_i\mathbf{g}+\sum_{j=1}^{M}k\boldsymbol{\xi}_{ij}+\mathbf{F}_i^{\text{col}} \tag{27}$$

and

$$I_i \frac{\mathrm{d}\omega_i}{\mathrm{d}t} = \mathbf{T}_i, \tag{28}$$

and the corresponding initial conditions for $\mathbf{U}_i$ and $\boldsymbol{\omega}_i$ are:

$$\mathbf{U}_i(t=0) = \mathbf{U}_{i,0} \quad \text{and} \quad \boldsymbol{\omega}_i(t=0) = \boldsymbol{\omega}_{i,0}. \tag{29}$$

Using these equations, the particle positions are updated by applying an explicit time-integral scheme.

It must be pointed out that the suspended particles move continuously in space and their boundary points are not confined to the lattice nodes as in the conventional LBM. Instead, once the velocities on these lattice nodes are calculated, we use the Dirac distribution function to interpolate the fluid velocity to the boundary points of the particles. In the simulations that follow, the Dirac distribution function given by Eq. (24) is used in the velocity interpolation scheme.

2.2. Collision technique

In any problem where a large number of particles flow in a confined volume, such as the sedimentation problem, collisions between particles are unavoidable. In this case it is of paramount importance to employ a numerical technique to deal with collisions in order to keep the computational particles from penetrating into each other's boundaries and the fluid container. Without a collision technique, the computational particles may overlap, thus violating the behavior of physical particles. There are several collision methods that are commonly used in particle-fluid interaction problems. One popular approach is to introduce a repulsive force when the gap between two particles exceeds a given threshold, the so-called safe zone. The repulsive force may be either a lubrication force [14] or a spring force [12]. The latter allows a small penetration between two particles and generates a strong repulsive force that pushes the two particles apart. For the simulation of the sedimentation problem presented here, we choose a collision technique that applies a repulsive force if the gap between the two particles is less than a given threshold. This extra short-range repulsive force is added as an external force into the total force that a particle experiences. The functional form of the repulsive force is as stipulated by Glowinski et al. [11]:

$$\mathbf{F}_{ij}^{\mathrm{P}} = \begin{cases} 0, & \|\mathbf{x}_i - \mathbf{x}_j\| > R_i + R_j + \zeta, \\ \frac{c_{ij}}{\varepsilon_{\mathrm{P}}} \left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\| - R_i - R_j - \zeta}{\zeta} \right)^2 \left(\frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \right), & \|\mathbf{x}_i - \mathbf{x}_j\| \leqslant R_i + R_j + \zeta. \end{cases} \tag{30}$$

The parameter, c_{ij}, is the force scale as defined above, which is chosen to be the buoyancy force on the particle for the sedimentation problems; ε_{P} is the stiffness parameter for collisions; R_i and R_j are the radii of two particles, respectively; and ζ is the threshold or "safe zone," which is specified in advance. Similarly the repulsive force between a particle and a wall is given by the reflection method as follows:

$$\mathbf{F}_{ij}^{\mathrm{W}} = \begin{cases} 0, & \|\mathbf{x}_i - \mathbf{x}_{i,j}\| > 2R_i + \zeta, \\ \frac{c_{ij}}{\varepsilon_{\mathrm{W}}} \left(\frac{\|\mathbf{x}_i - \mathbf{x}_{i,j}\| - 2R_i - \zeta}{\zeta} \right)^2 \left(\frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \right), & \|\mathbf{x}_i - \mathbf{x}_{i,j}\| \leqslant 2R_i + \zeta, \end{cases} \tag{31}$$

where $\mathbf{x}_{i,j}$ is the position of a fictitious particle $P_{i,j}$, which is located symmetrically on the other side of the wall W_j, and ε_{W} is another stiffness parameter. Glowinski et al. [11] provided the justification and an extensive discussion on how to choose these stiffness scale parameters.

Based on the above set of governing and closure equations, the computational scheme that has been used for the implementation of the IB-LBM in the particle sedimentation problem may be summarized as follows:

(1) At $t = t_n$, the fluid velocity and particle velocity (translational and rotational), together with the particle position (initial position when $t = 0$) are known.
(2) The positions of particle reference points are computed using the particle rigid body motion equations. The velocity on each particle boundary point is obtained through the interpolation of the fluid velocities on neighboring nodes.
(3) The deformations of the particles' boundary are calculated and the internal link force on each boundary point is computed according to Eq. (17).
(4) The collision rule is applied and repulsive forces are added if the gap between two particles or one particle and a wall is less than the prescribed threshold.
(5) The force on boundary points is transferred into the lattice nodes. The summation of all these transferred forces yields the body force density.
(6) The flow field is solved using the LBM.
(7) The internal link forces together with the external forces are used to update the translational and rotational velocity of the particles.
(8) The particles' position is updated.

It must be pointed out that, as in the case of the bounce-back rule that treats the particle–fluid interaction in LBM, the rigid body motion inside the particle is not a priori enforced. Therefore, the problem solved is actually the interaction between a fluid and a solid shell. The shell has the same boundary as the particle does and carries the total mass of the particle, but the contribution of the particle interior is ignored. It is fortuitous in this case that the effect of the interior fluid is not significant. This was proven by Ladd [15] who showed that accurate simulations could be carried out with or without considering the interior fluid.

3. The flow of a neutrally buoyant particle in linear shear flow

We have applied the IB-LBM to investigate the motion of a neutrally buoyant two-dimensional circular particle moving in a viscous fluid. This problem has been extensively studied by others, notably by Feng et al. [3] using a finite element method as well as by Feng and Michaelides [5,6] using the LBM. Therefore, the comparison of the results of the IB-LBM with results from other numerical methods will serve as a validation of the method.

A schematic diagram of the problem solved is depicted in Fig. 3, and the parameters used in the simulation are as follows: the radius of the particle $a = 10$; the gap between two plates $H = 80$; the plate width $L = 2000$; the relaxation time $\tau = 0.6$, which yields for the fluid kinematic viscosity in lattice units $\nu = 1/30$;

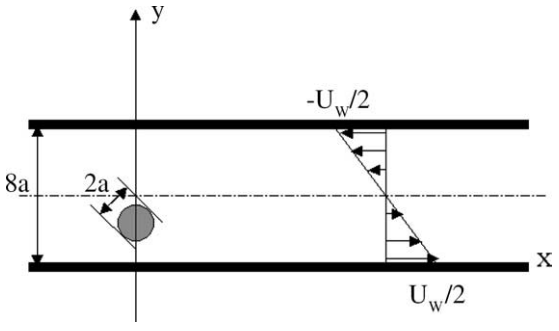

Fig. 3. Migration of a neutrally buoyant particle in a simple shear flow between two walls.

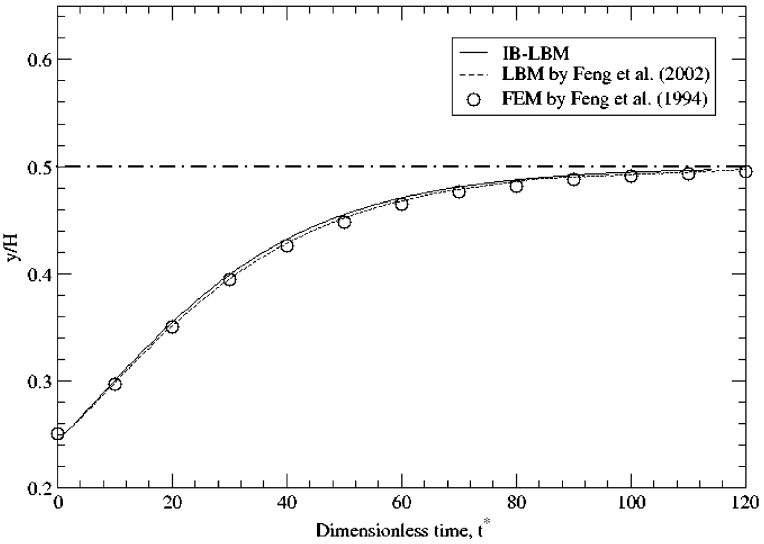

Fig. 4. A comparison of the results of this study with results obtained by others using the LBM and FEM.

the spring constant $k = 2.0$. Both upward and downward walls are moving at velocities equal to $U_W/2 = 1/120$ in opposite directions. This gives a shear rate for the flow $\gamma = 1/4800$. Based on this, the fluid bulk Reynolds number is $Re_b = U_W H/v = 40$ and the particle Reynolds number $R_P = \gamma * a^2/v = 0.625$. The particle is initially at the position $y_0 = 0.25H$ above the bottom wall or halfway towards the center and is initially at rest. This initial position results in a local slip velocity between the particle and fluid equal to $-0.25U_W$. The lateral migration of the particle using the current IB-LBM is compared with the results from the LBM and the FEM in Fig. 4. It is shown that there is a very good agreement between the results obtained from the three different numerical methods.

Fig. 5 also shows the two components of the particle translational velocity obtained by the IB-IBM and LBM. An almost perfect agreement is observed with the results obtained from the two methods. Considering that now the LBM is a well-established method for the solution of fluid–particle interaction

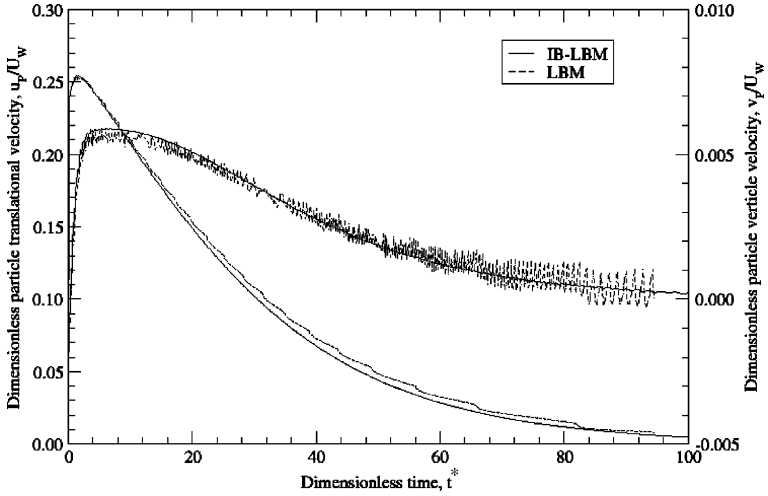

Fig. 5. Particle translational velocities calculated by the IB-LBM and LBM.

problems, the results shown in Figs. 4 and 5 are a validation of the accuracy of the proposed method. It is also observed in Fig. 5 that the velocities calculated by IB-LBM are smoother than those obtained by LBM. This is an advantage of the IB-LBM and is due to the absence of variability of the computational boundary, which was discussed in the previous section and pictorially demonstrated in Fig. 1.

4. Sedimentation of two circular particles in a viscous fluid

4.1. Results by IB-LBM

In order to demonstrate the use of the IB-LBM and further validate its results, we simulated the sedimentation of two circular particles in a channel and, essentially, replicated the problem solved by Patankar et al. [18] and Patankar [17] who used the fictitious domain method. The channel is 2-cm wide (x-direction) and 8-cm high (y-direction). The fluid has the properties of water with viscosity 0.001 g/cm s and density 1 g/cm^3. The particles density is 1.01 g/cm^3, and the radii of the particles are 0.1 cm. Initially, the first particle is 0.001 cm off the channel center at a height of 7.2 cm and the second particle is at the channel center at a height 6.8 cm. Both particles and fluid are initially at zero velocity. The two particles start settling in the y-direction, due to the gravity force. It is well known that these two particles will undergo a motion called "drafting, kissing and tumbling" or DKT, which was first numerically demonstrated by Feng et al. [2].

For the solution of this problem, we used a computational domain of 200×800 lattice units. Each particle was covered by 20 lattice units and the relaxation time is $\tau = 0.65$. Thus, each lattice simulation step corresponds to 5.0×10^{-4} s. The stiffness scale factor ε_d of the boundary was chosen to be 0.25, and 160 boundary nodes were used. The repulsive force was added when the gap between two particles was less than 1 lattice unit and when the gap between one particle and the wall was less than half of one lattice unit. For the particles collision, the stiffness coefficient ε_P was 2.0.

Fig. 6 shows the positions of two particles at seven different times. It must be pointed out that the DKT motion that is observed in Fig. 6 was also observed by Patankar et al. [18] and by Patankar [17] who used different methods.

The velocity vector maps during several instances of the simulation at times $t = 1.5$, 2.5, 3.0, and 4.5 s are presented in Fig. 7, and the instantaneous transverse coordinates of the centers of the two particles are depicted in Fig. 8.

It must be pointed out that, even though the transverse position of the two particles is the same, the particles do not overlap because their instantaneous longitudinal positions are different. This is clearly depicted in Fig. 9, which shows the longitudinal component of the trajectories of the two particles.

Figs. 10 and 11 depict the instantaneous velocity components of the two particles in the transverse and the longitudinal directions.

It is seen in Figs. 6–11 that initially the first particle (particle 1) trails the other particle (particle 2) by about 0.4 cm. The two particles keep a steady distance up to about $t = 0.8$ s, when the particles start to approach closer. At approximately $t = 1.4$ s, the distance between these two particles is about one particle diameter, which implies that the inter-particle forces are significant (the "kissing" part of the motion). At the same time, the centers of the two particles start to slowly deviate from the channel center and then, the particles completely tumble at time $t = 2.4$. During the "tumbling" part of the process, the two particles move closer to each other, as it is shown by their longitudinal velocity components in Fig. 11.

It must be pointed out that we also run a simulation with a shorter time step, 2×10^{-4} s, and found nearly identical results.

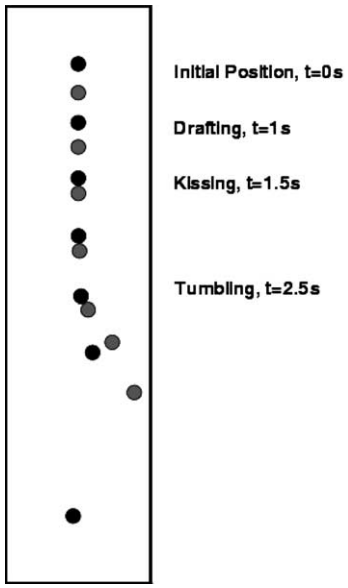

Fig. 6. Sedimentation of two circular particles in a channel at different time stages.

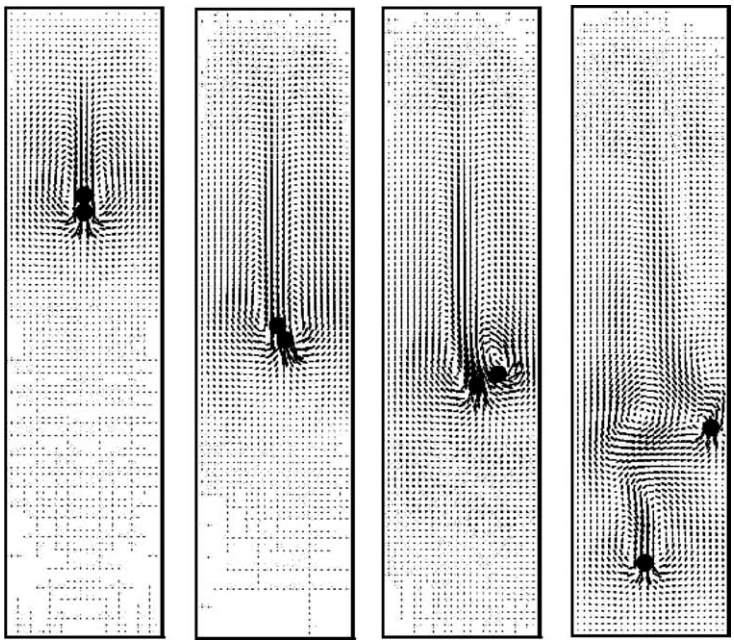

Fig. 7. Flow velocity vector maps at: (a) $t = 1.5$ s; (b) $t = 2.5$ s; (c) $t = 3.0$ s; and (d) $t = 4.5$ s.

Regarding the parameters used in the collision technique as described by Eqs. (30) and (31), we found that the difference of the computational results is insignificant when we choose the "safe zone" between the colliding particles to be between 1 and 0.5 lattice units. Also, when the stiffness factor is in the range

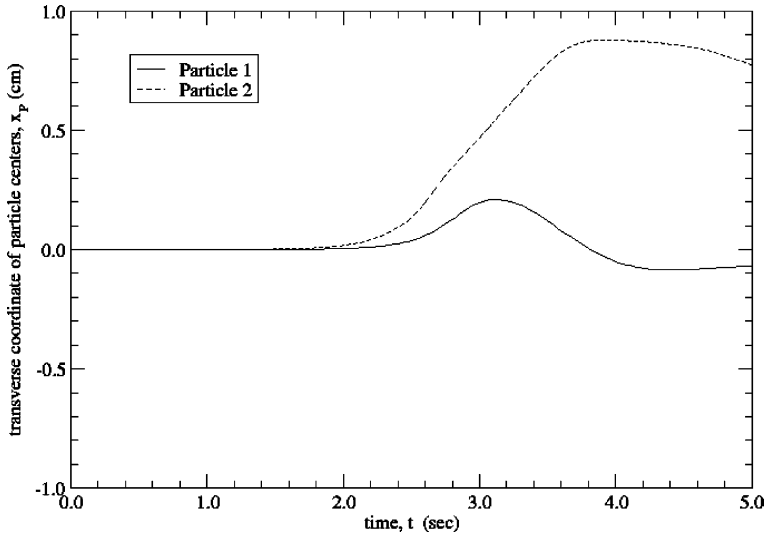

Fig. 8. Transverse coordinates of the centers of the two particles.

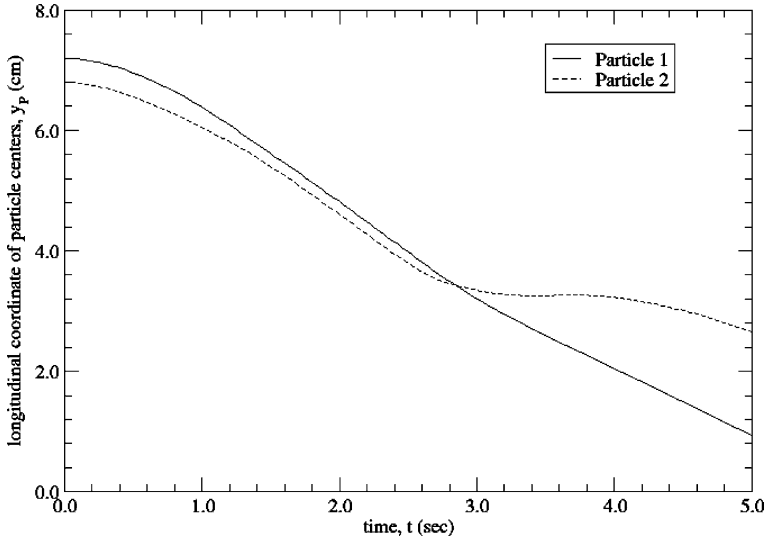

Fig. 9. Longitudinal coordinates of the two centers.

$1 < \varepsilon_P < 10$, the results obtained are almost identical. However, we observed a significant change in the results when the collision stiffness factor ε_P was increased drastically from 2 to 50, which is far from the recommended range. The main effect on the velocities is during the "kissing" and "tumbling" part of the process. The observed differences are depicted in Fig. 12, where it may be seen that the results obtained with the very high value of the collision stiffness factor at $t > 3$ differ significantly.

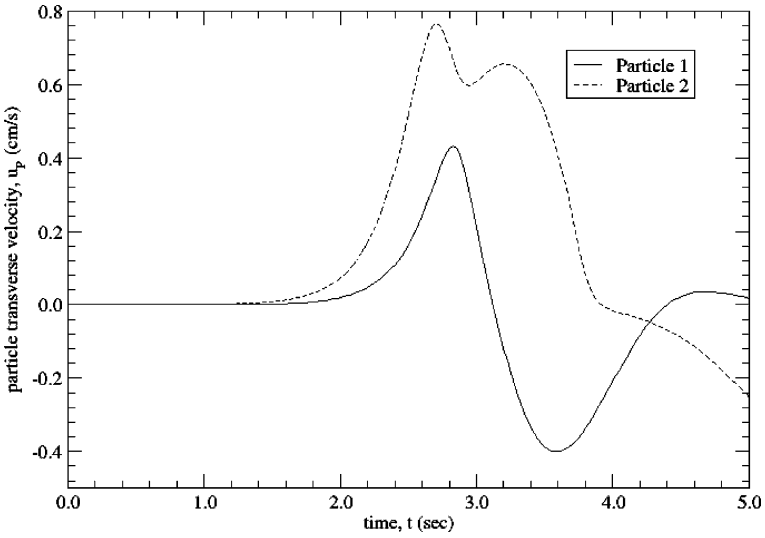

Fig. 10. Transverse velocities of the two particles.

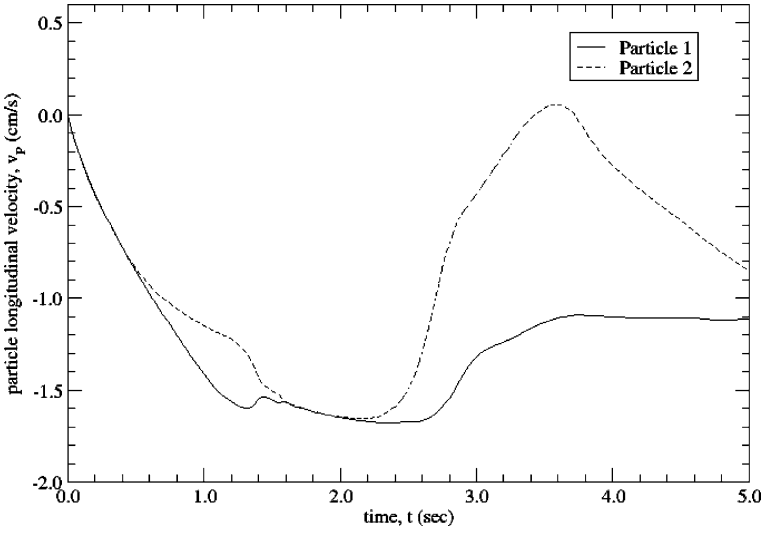

Fig. 11. Longitudinal velocity components of the two particles.

4.2. Sensitivity to the surface stiffness scale factor ε_d for the particle boundary

As long as the deformation of the particle is kept small, the results of the proposed method are not sensitive to the choice of the stiffness scale factor ε_d of the particle boundary. In order to investigate the sensitivity of the results on ε_d, we performed computations and simulated the DKT process with several different values of the stiffness scale factor, ε_d Besides using the recommended value, $\varepsilon_d = 0.25$, we also calculated the results at stiffness scale factor $\varepsilon_d = 2.5$, $\varepsilon_d = 0.5$, and $\varepsilon_d = 0.17$, respectively. Fig. 13 shows the y-components of the particle velocities with these parameters. It is seen that the results at $\varepsilon_d = 0.25$ and

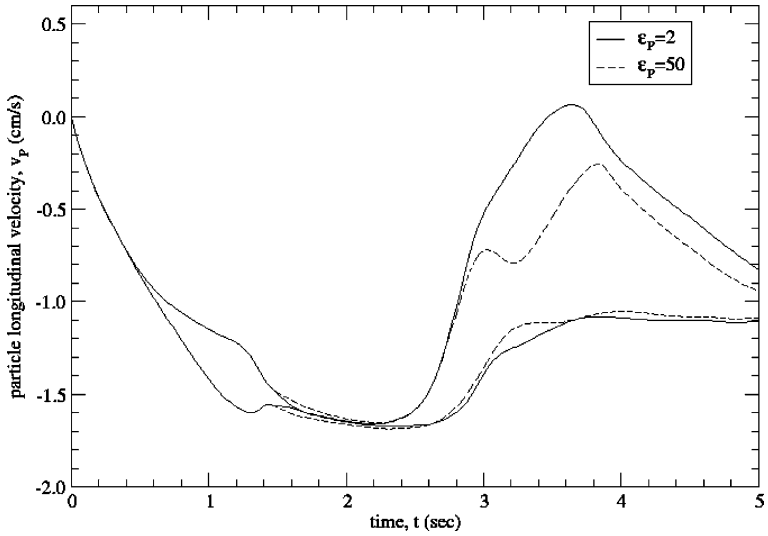

Fig. 12. Longitudinal velocities using two drastically different collision stiffness factors, ε_P.

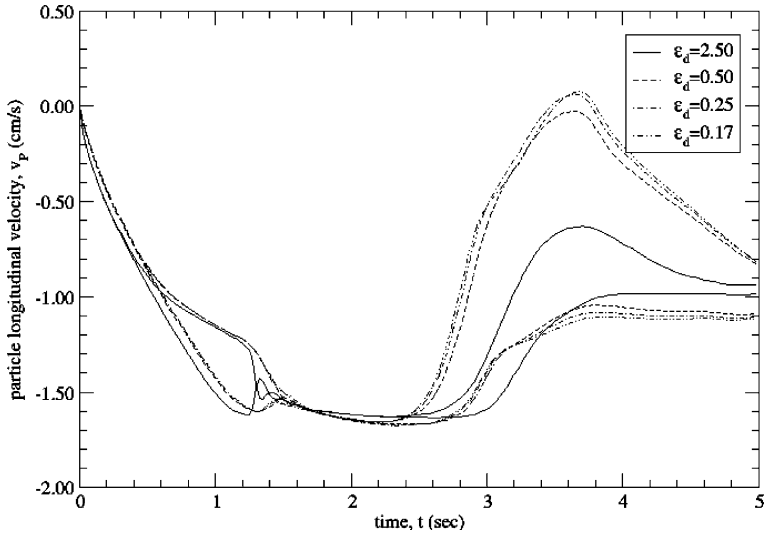

Fig. 13. Longitudinal velocities using different stiffness scale ε_d.

$\varepsilon_d = 0.17$ are almost identical and there is little difference of the results even when $\varepsilon_d \leqslant 0.5$. However, there are significant differences when the highest value, $\varepsilon_d = 2.5$ was used. From the results of Fig. 13, one may conclude that there is no significant difference in the computation of the velocities of the particles as long as $\varepsilon_d \leqslant 0.5$. Also, that when the stiffness scale factor is large, $\varepsilon_d = 2.5$, the boundaries of the particles become too soft, the particle deformation becomes significant and affects the flow.

In order to demonstrate that this is due to the deformation of particles, we also computed the maximum segment deformation ratio of the two particles surface segments during the DKT process for the four cases

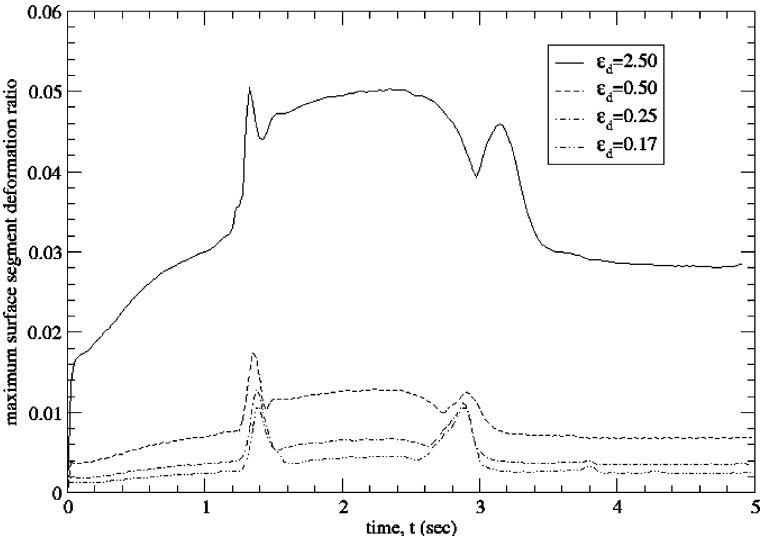

Fig. 14. Maximum deformation ratio during DKT process.

depicted in Fig. 13. The segment deformation ratio is defined as the change of the segment length divided by its original length. The maximum segment deformation ratio is the maximum value among the entire particles' segment deformation ratio and is depicted in Fig. 14. As it can be seen in this figure, the maximum deformation for the DKT process occurred around $t = 1.4$ s, which is the time when two particles start the "kissing" and "tumbling" part of the process. The maximum deformation ratio during the entire DKT process using the four different stiffness scale factors is also presented in Table 1. From this table and Fig. 14, it is seen that at $\varepsilon_d = 2.5$, the deformation ratio is as large as 5%, and overall, it remains between 3% and 5%, while in the other cases the deformation ratio is in general below 1%. From this, we may conclude that in order to achieve satisfactory results, the maximum deformation ratio should be kept less than 2% and, probably close to 1%.

4.3. Grid size effect

The results obtained in the simulations mentioned in the previous sections were obtained using a computational domain of 200×800, with the diameter of each particle being 20 lattice units, $d = 20$. Therefore, the channel has 200 transverse grid points, and the physical grid step is 0.01 cm. To demonstrate the affect of grid size to the computational results, we use two different grids, while keeping the grid size to diameter ratio constant. Thus, we use a coarser grid of 100×400 with the particle diameter being equal to 10 lattice units as well as a finer grid of 300×1200 with the particle diameter being 30 lattice units. We performed computations with these computational grids and show some of the results of the computations for the longitudinal velocities in Fig. 15.

Table 1
The maximum deformation ratio among surface segments during entire DKT process for different ε_d

Stiffness scale factor, ε_d	2.50	0.50	0.25	0.17
Maximum deformation ratio (%)	5.1	1.8	1.3	1.0

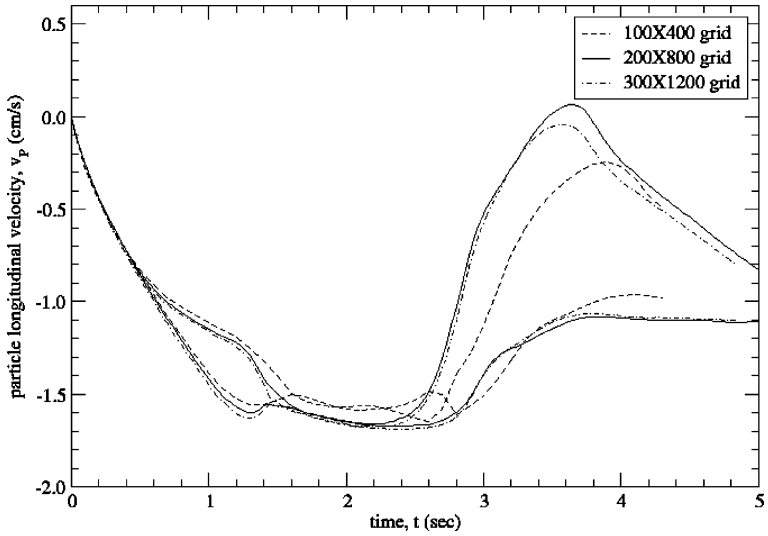

Fig. 15. Particle longitudinal velocities with different grid.

It is observed that the results using the 200×800 and 300×1200 grids are almost identical up to $t = 3.5$ s and diverge only by a small amount thereafter. Even with the coarser grid of 100×400, the results obtained show good qualitative and quantitative agreement with those obtained using the finer grids.

4.4. Comparison with results of simple LBM

The LBM is now a mature computational method that has been validated by many researchers. To compare with results from the IB-LBM, we solved the same DKT problem using the simple LBM. In general, we have found that the CPU time of the IB-LBM and LBM is comparable for this problem. For example, in this case the LBM took 43 min of CPU time and the IB-LBM took 45 min of CPU time in a SGI Onyx 3400 machine. Figs. 16 and 17 depict the results of the particle transverse and longitudinal velocities obtained by these two methods.

As seen in Figs. 16 and 17, the velocities calculated by the two methods agree very well before the tumbling part of the process. The tumbling process lasts approximately 0.9 s in the IB-LBM computations, while it in the LBM case it is much shorter and lasts only 0.4 s. Numerical results obtained by others [17,21] indicate that the tumbling part of the process lasts longer, between 0.8 and 1.1 s. This range includes the duration of the process obtained by the IB-LBM, a fact that tends to validate the results of the method. It is believed that this difference is due to the simulations of particles interaction. In this case the particle interaction is the result of instability and the imperfections of the particle surface, which are inherent in the LBM, tend to accentuate the effects of this instability as also noted by Zhang and Prosperetti [21]. It has been demonstrated in the previous section that the description of particle boundary is different in the LBM and IB-LBM and that the latter provides a more accurate approach to handle the smoothness of the particle boundary. Actually, the results of FDM that will be discussed in the following section also show a similar tumbling time period as the one calculated by IB-LBM. This is an indication that the IB-LBM treats the particle boundary better than the LBM does. One may also conclude that for the fluid–particle interaction problems, the actual results are affected considerably by the way each numerical method handles the particle surfaces and the particles collisions and it is evident that the IB-LBM does a good job in handling the particle boundary.

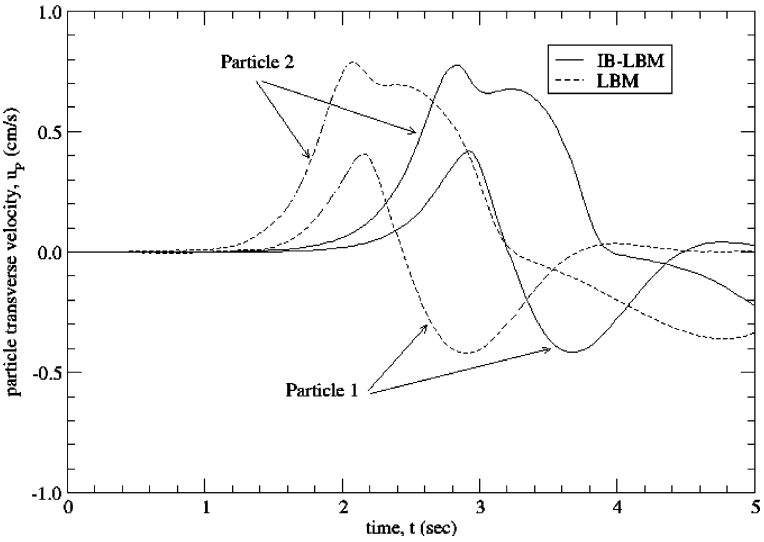

Fig. 16. Particle transverse velocities obtained by the IB-LBM and the LBM.

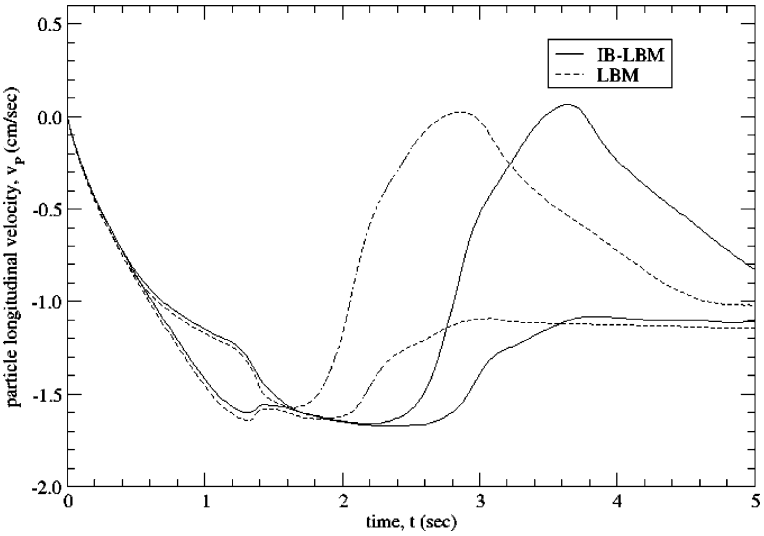

Fig. 17. Particle longitudinal velocities calculated by IB-LBM and LBM.

4.5. Comparison with results from FDM

The whole DKT process obtained by using the IB-LBM agrees qualitatively very well with the results of the same process obtained by Patankar et al. [18] and Patankar [17]. In order to make a quantitatively comparison, we plotted the longitudinal velocities obtained by using the IB-LBM (using the 200×800 grid) together with the results obtained from Patankar et al. [18] and Patankar [17] using the FDM. This comparison is shown in Fig. 18.

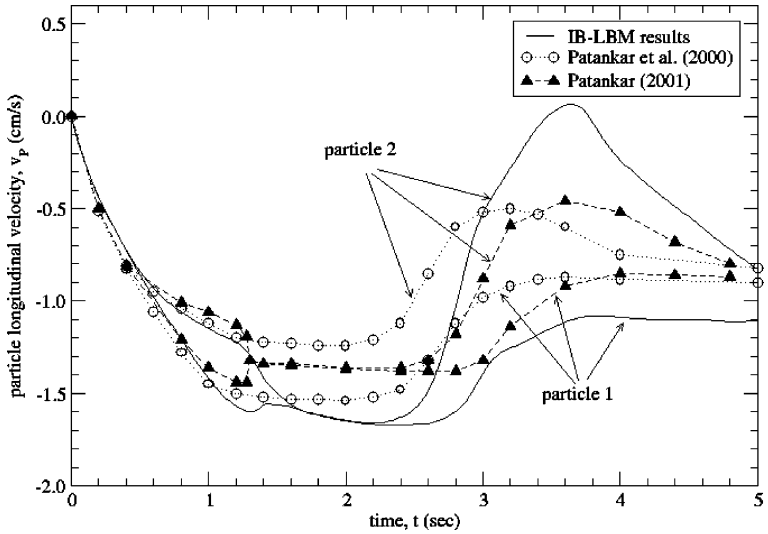

Fig. 18. A comparison of the results of this study with those obtained by Patankar et al. [18] and Patankar [17] using the FDM.

Ii is observed that before the two particles start the "kissing and tumbling" part of the process, there is a very good agreement of the results obtained by the IB-LBM and the those obtained by the other studies. Thereafter all results diverge by a small amount. However, it is also observed that the results obtained by Patankar and his colleagues in the two studies also differ quantitatively. Patankar [17] has explained this apparent disagreement as follows: the tumbling part of the process, which is an instability, is initiated at slightly different stages in the two simulations and this causes different velocity histories thereafter. This occurs because the DTK process is unstable and such a problem "is inherent in these simulations, since the tumbling process is a realization of an instability and can be affected by the accuracy of the solution procedure and modeling of the collision forces" [17].

5. Sedimentation of a large number of circular particles in an enclosure

This problem has been studied by Glowinski et al. [10] by using the FDM. The initial setup of the problem is shown in Fig. 19. A large number (504) of circular particles is enclosed in a closed two-dimensional box. The box has 2 cm width and 2 cm height, and the diameter of each one of the circular particles is $d = 0.0625$ cm. The fluid density is $\rho_f = 1$ g/cm^3, and the particle-to-fluid density ratio is 1.01. The fluid kinematic viscosity is 1 g/ms. Because of the large number of particles, the range over which the repulsive force is enabled is $d/16$. For the computations, the following values were used for the stiffness parameters of the collisions: $\varepsilon_P = 2$ and $\varepsilon_W = 0.5\varepsilon_P$.

Initially there are 18 lines of particles with each line having 28 particles. The gap between any two neighboring particles is $2d/16$. The gap between the upper wall and the first horizontal line (Line 1) is $6d/16$. The gap between the left wall with the leftmost particle in all the odd-numbered horizontal lines, that is lines $1, 3, 5, \ldots$, is $6d/16$. The gap between the left wall with the leftmost particle in all the even horizontal lines, that is lines $2, 4, 5, \ldots$, is $4d/16$. At time $t = 0$, both the fluid and particles are stationary with the particles being on top of the fluid.

The simulation box used in the IB-LBM simulations is composed by 512×512 lattice nodes and each particle is covered by 16 nodes. The relaxation time in this case is $\tau = 0.9915$, and every time step

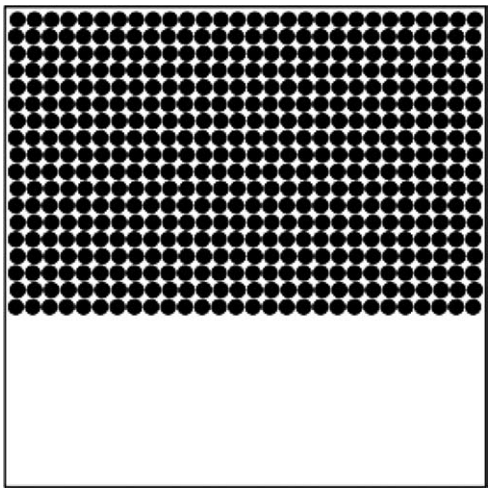

Fig. 19. Initial positions of the 504 particles in an enclosure.

corresponds to a physical time of 0.00025 s. The surface stiffness scale factor ε_d is 0.5. The results of the simulation of this problem, which is essentially a Rayleigh–Taylor instability, are depicted in several snapshots in Figs. 20–22. It is seen in Fig. 20 that initially all the particles start settling uniformly. The effect of the walls in hindering the closest particles is clearly observed in Figs. 21 and 22, leading to the creation of two eddies close to the sidewalls. When these two eddies run into the sidewalls, each eddy splits into two smaller eddies, one moving upwards and the second one moving downwards, as depicted in Fig. 23. Subsequently, the two downward eddies, one at each side of the container, grow and start to pull particles downwards, as depicted in Fig. 24. At this stage of the process, these two eddies are the dominant features of the flow in the lower half of the enclosure. They appear to be strong enough to be able to pick up particles and shoot them upwards, as it can be seen in Fig. 25, which is the snapshot towards the end of the

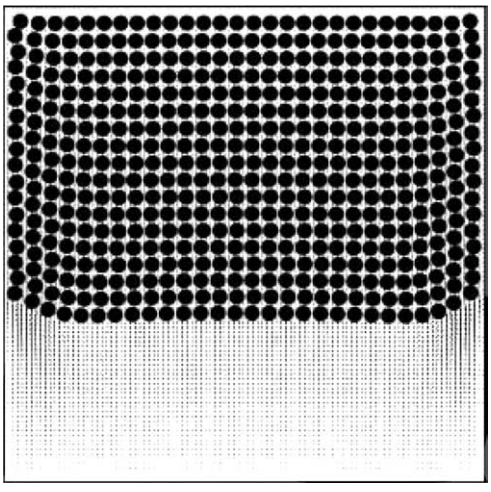

Fig. 20. Positions of the 504 particles at $t = 1$ s.

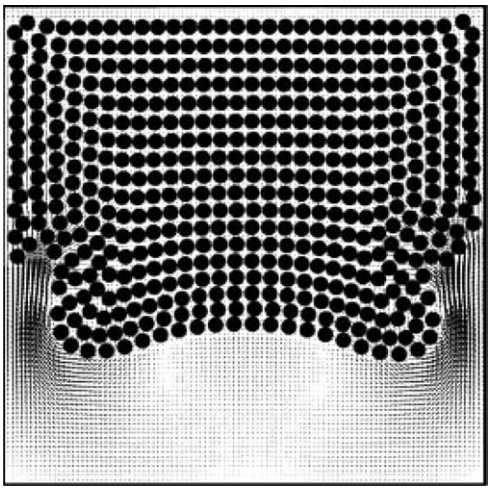

Fig. 21. Position of the 504 particles at $t = 2$ s.

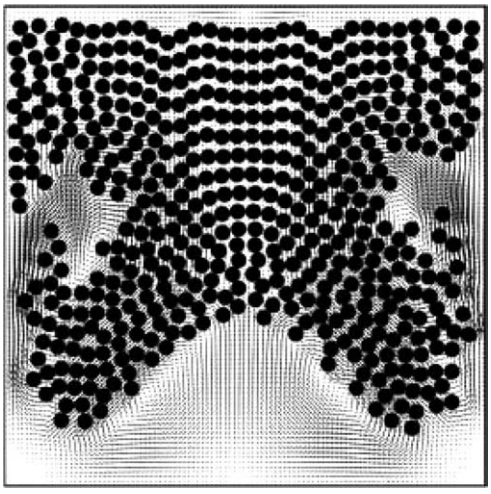

Fig. 22. Positions of the 504 particles at time $t = 3$ s.

process. Figs. 26 and 27 show the final stages of the settling process, when all the particles have settled in the bottom of the container. There is some ''packing'' of the particles in the last two stages as it becomes apparent by looking at the bottom row of the two figures, where two small gaps appear in Fig. 27, even though the total number of particles on the row is the same in the two cases. These two gaps are an indication of the slow process particle packing, which is expected to conclude the sedimentation process.

The patterns observed in this simulation are very close to the results given in the paper by Glowinski et al. [10] up to $t = 3$ s. The details of some particles' positions differ at $t > 3$ s, but the flow patterns observed are always the same. The discrepancy is most probably due to the method that accounts for the instabilities and the collisions of the particles. This is more pronounced at times greater than $t = 3$ s when the instabilities and collisions of particles are much more frequent. However, the faithful reproduction of

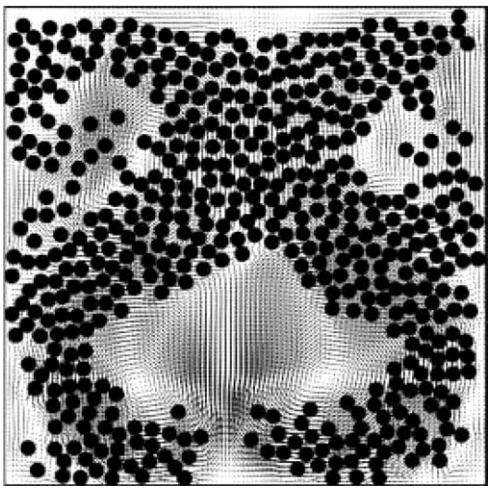

Fig. 23. Positions of the 504 particles at time $t = 4$ s.

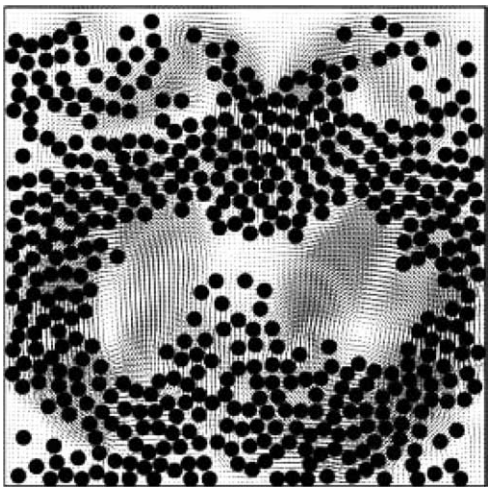

Fig. 24. Positions of the 504 particles at $t = 5$ s.

the details of the process, including the creation and growth of the two eddies at the sides of the container, is a good indication of the validity and usefulness of the method used for the simulation.

It must be pointed out that the present simulations are mainly conducted on a SGI Onyx 3400 machine at the Center for Computational Science at Tulane University. In the case of DKT stimulation with 200×800 grid, it took 45 min to simulate 5 s physical time by using the IB-LBM. In the case of sedimentation with 504 particles, the time to complete one iteration is about 2.8 s, which results in approximately 3.1 h computational time to simulate 1 s of physical time.

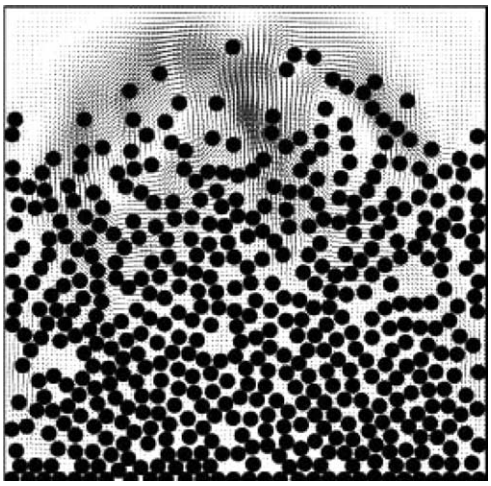

Fig. 25. Positions of the 504 particles at time $t = 8$ s.

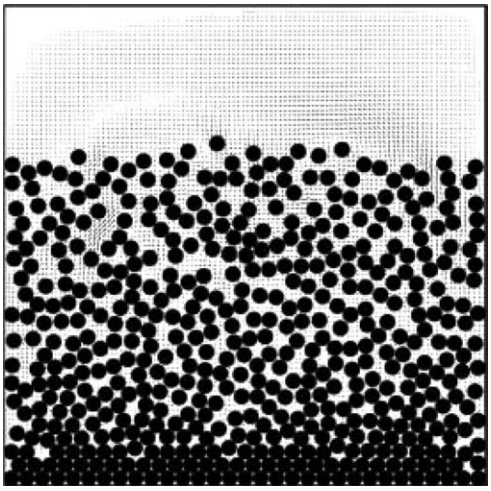

Fig. 26. Positions of the 504 particles at time $t = 12$ s.

6. Conclusions

A new method, called IB-LBM, which combines desired features of the Lattice Boltzmann Method and the Immersed Boundary Method has been developed. This method utilizes two computational grids, one for the flow domain, which is Eulerian, and the second for the particles in the flow, which is Lagrangian. The rigid body conditions are enforced by the penalty method assuming the particle boundary to be deformable with a high stiffness constant. The constraint force is calculated through the displacement of the boundary point (tracer) and reference point (mark). The force density, due to the constraint force, is added into the lattice Boltzmann equation, and the composite velocity field of the fluid and particles is obtained after solving the lattice Boltzmann equation. This new method preserves the advantages of LBM in dealing

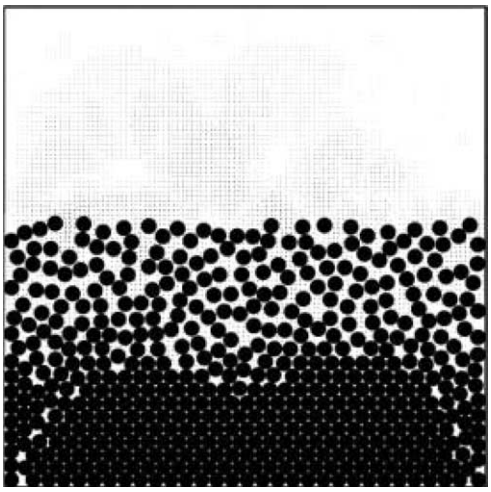

Fig. 27. Positions of the 504 particles at time $t = 24$ s.

with multiple particles and also provides an alternative and accurate approach to treating the solid–fluid boundary conditions. In addition, the IB-LBM is capable to easily simulate flows with deformable particles and solid-structure deformation problems.

The IB-LBM has been validated by comparing the results of a neutrally buoyant particle migration in a linear shear flow with FEM and LBM, and by comparing its results with the results from previous simulations on the DKT process of two circular particles and the sedimentation process of 504 particles in an enclosure. The simulations have proven that the IB-LBM is a robust method that is capable to solve particle–fluid interaction problems with a large number of particles. Compared with the traditional LBM, the IB-LBM provides a more accurate approach than the "bounce-back rule" for the treatment of the solid–fluid boundary. It is found through our numerical experiments that the intermediate results, such as particle forces and torques, do not exhibit the initial fluctuations (noise) that are common in the traditional LBM. As for the efficiency of the computations, the present method falls slightly behind the traditional LBM, because of the use of a set of Lagangian points on the particle boundary. However, instead of using step-wise links between lattice nodes, the IB-LBM represents the particle boundary in a smoother way. In addition, the computational boundaries for the moving particles do not vary in every time-step. This avoids the fluctuation of the particle forces and velocities that were observed in simulations using the unaltered LBM. Most importantly, the present method is capable of solving structure-deformation problems that the traditional LBM cannot do.

Finally, it must be pointed out that even though the simulations presented here are limited to two-dimensional flows, the equations presented are general and the IB-LBM may be easily expanded, with some modifications, to three-dimensional flows because the concept of the immersed boundary is still valid in three-dimensional flows.

Acknowledgements

We are thankful to Dr. X.D. Wang (Polytechnic University of New York) for several insightful discussions on the Immersed Boundary Method. This research was partly supported by two grants from the

USGS and the DOE to the Tulane-Xavier Center for Bioenvironmental Research. Computational resources were provided by the Tulane/Xavier Millennium Center for Computational Studies.

References

[1] J.F. Brady, G. Bossis, Stokesian dynamics, Ann. Rev. Fluid. Mech. 20 (1988) 111–157.
[2] J. Feng, H.H. Hu, D.D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 1. Sedimentation, J. Fluid Mech. 261 (1994) 95–134.
[3] J. Feng, H.H. Hu, D.D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 2. Couette and Poiseuille flows, J. Fluid. Mech. 277 (1994) 271–301.
[4] J.M. Buick, C.A. Greated, Gravity in a lattice Boltzmann model, Phys. Rev. E (2000) 5307–5320.
[5] Z-G Feng, E.E. Michaelides, Hydrodynamic force on spheres in cylindrical and prismatic enclosures, Int. J. Multiphase Flow 28 (2002) 479–496.
[6] Z-G Feng, E.E. Michaelides, Interparticle forces and lift on a particle attached to a solid boundary in suspension flow, Phys. Fluids 24 (2002) 49–60.
[7] A.L. Fogelson, C.S. Peskin, A fast numerical method for solving the three-dimensional Stokes equation in the presence of suspended particles, J. Comput. Phys. 79 (1988) 50–69.
[8] U. Frisch, B. Hasslacher, Y. Pomeau, Lattice-gas automata for the Navier–Stokes equations, Phys. Rev. Lett. 56 (1986) 1505.
[9] U. Frisch, D. D'Humirés, B. Hasslacher, P. Lallemand, Y. Pomeau, J.P. Rivert, Lattice-gas hydrodynamics in two and three dimensions, Complex Syst. 1 (1987) 649–707.
[10] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 25 (1999) 755–794.
[11] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, J. Comput. Phys. 169 (2001) 363–426.
[12] K. Höfler, S. Schwarzer, Navier–Stokes simulation with constraint forces: finite-difference method for particle-laden flows and complex geometries, Phys. Rev. E 61 (2000) 7146–7160.
[13] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation Part I. Theoretical foundation, J. Fluid. Mech. 271 (1994) 285–310.
[14] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part II. Numerical results, J. Fluid Mech. 271 (1994) 311–339.
[15] A.J.C. Ladd, R. Verberg, Lattice-Boltzmann simulation of particle–fluid suspensions, J. Stat. Phys. 104 (2001) 1191–1251.
[16] E.E. Michaelides, Freeman scholar paper – hydrodynamic force and heat/mass transfer from particles, bubbles and drops, J. Fluids Eng. 125 (2003) 209–238.
[17] N.A. Patankar, A formulation for fast computations of rigid particulate flows, Annual Research Briefs-2001, Center for Turbulence Research, Stanford University, 2001.
[18] N.A. Patankar, P. Singh, D.D. Joseph, R. Glowinski, T.-W. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 26 (2000) 1509–1524.
[19] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[20] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[21] Z. Zhang, A. Prosperetti, A method for particle simulation, J. Appl. Mech., Trans. ASME 70 (2003) 64–74.