Computers & Fluids 118 (2015) 159–166

Contents lists available at ScienceDirect

Computers & Fluids

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

A hybrid FD-DEM solver for rigid particles in viscous fluid

Shifeng Wu [a,b,*], Li Yuan [b]

[a] Department of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, PR China
[b] LSEC and Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, PR China

ARTICLE INFO

Article history:
Received 26 December 2013
Received in revised form 28 April 2015
Accepted 3 May 2015
Available online 8 June 2015

Keywords:
Collision
Particulate flow
Repulsive force
Fictitious domain method
Discrete element method
Discrete δ-function

ABSTRACT

In this article, the dynamics of colliding particles immersed in a viscous fluid is numerically investigated. An improved fictitious domain (FD) method for a solid–fluid system is developed and a collision strategy for multiple particles is presented. In earlier methods, a repulsive force is applied to the particles when their separation distance is less than a critical value and depending on the magnitude of this artificial repulsive force, collision of two or more particles may bounce unrealistically. In the present method, when the collision of two or more particles happens, the collision force acting on each colliding particle is determined by using the Discrete Element Method (DEM). Furthermore, the improved FD method employs a discrete δ-function in the form of bi-function to transfer quantities between the Eulerian and Lagrangian nodes similar to the immersed boundary method. This feature avoids the need to use Lagrange multipliers for imposition of the rigid body motion. Numerical results for motions of one or several particles in a viscous fluid are presented.

1. Introduction

In order to accurately predict the behavior of particulate flows, a fundamental knowledge of the dynamics of particle collision is required. Classical lubrication theory predicts that the lubrication force becomes singular as the distance between two smooth spheres approaches zero and hence prevent spheres from touching. In practice, no general numerical method can afford the computational cost of resolving the flow in the narrow gaps between closely-spaced particles and therefore some modeling is needed. Physically, the surface of particles has some roughness, and the bumps make physical contact possible due to the discrete molecular nature of the fluid and/or the attractive London-van der Waals forces. Thus further approach is prevented as the solid–solid contact occurs [1]. Many numerical methods adopt a heuristic repulse force the solid–solid contact [1,10,12,14].

Historically, the non-boundary-fitted (or Cartesian grid) method is a popular choice for the numerical solution of incompressible fluid flow problems in complex geometries or with moving boundaries. A variety of non-boundary-fitted approaches have been developed, and they can be roughly classified into two families [2,3,13]: the body-force based method (e.g., [4]) and the non-body-force based method (e.g., [2]). For the former, a body force (or momentum forcing) is introduced into the momentum equations. The fractional step methods are often used in the body-force based method to simplify the computation in the following way: the Navier–Stokes equations are solved for the known body-force obtained at the previous fractional steps, with the boundary condition on the immersed boundary disregarded, and the boundary condition is used to determine the body-force at the subsequent fractional steps. There exist various body-force based methods in the literature that differ in the way how the body-force is calculated. Unlike the body-force based method, the non-body-force based method such as the immersed interface method (IIM) does not introduce a body-force, instead, the method (like IIM) accounts for the boundary condition by either transforming it into independent equations [5] or using it to modify the expressions of differential operators for the Eulerian nodes in the immediate vicinity of the boundary [6,7]. One advantage of the non-body-force based method is that the jump boundary conditions on the surface can be handled more accurately. But its disadvantage is the increased complexity.

We are concerned with numerical simulation of particulate flows in the present study. In principle, all aforementioned methods can be applied to calculating particulate flows, however, the body-force based method has been predominantly used so far. We also follow this line and focus on one type of such methods-the fictitious domain (FD) method. A common feature for the body-force based method is that the hydrodynamic force acting on the particles can be calculated from the body-force

* Corresponding author at: Department of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, PR China.
E-mail address: shifengwu@lsec.cc.ac.cn (S. Wu).

http://dx.doi.org/10.1016/j.compfluid.2015.05.028
0045-7930/

without integrating the pressure and shear stresses acting on the particle surface. For the distributed-Lagrange-multiplier/fictitious-domain (DLM/FD) method proposed by Glowinski et al., the particle velocities and the body-force (Lagrange multiplier) are solved simultaneously [4,11] with the implicit scheme, while for the immersed boundary method, they are obtained explicitly. The DLM/FD method has been successfully applied to a wide range of particulate flow problems [9]. However, its calculation procedure and iterative determination of the particle velocity and body-force is more involved and more expensive than the direct-forcing immersed boundary (IB) method. At the same time, most previous DLM/FD methods do not allow direct contact when two particles are closer. The aim of the present study is to present a simpler non-Lagrange-multiplier version of FD method without sacrificing the accuracy by employing a discrete δ-function in the form of bi-function to transfer explicitly quantities between the Eulerian grids and the Lagrangian nodes, as done as in the immersed boundary method. Also, we used a collision strategy of discrete element method (DEM), in which no repulsive force model is applied to the particles, rather, the contact force between particles is computed by using the DEM. Compared with the IB method, the particle velocity in the present method is not updated explicitly thus preventing numerical instabilities often encountered in IB method.

This paper is arranged as follows. A mathematical formulation of the basic scheme is first described. Numerical implementation of the scheme in a staggered grid, finite-volume framework is provided next. The motion of freely falling circular particles at different Reynolds numbers is simulated and the results are compared with previous numerical studies. Concluding remarks are given in the final section.

2. Numerical method

2.1. Theoretical formulation

For particulate flows, the governing equations in the fluid domain are

$$\rho_f \frac{d\mathbf{u}}{dt} = \nabla \cdot \boldsymbol{\sigma} \quad \text{in } \Omega_f, \tag{1a}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_f, \tag{1b}$$

$$\mathbf{u} = \mathbf{U} + \boldsymbol{\omega} \times \mathbf{r} \quad \text{on } \partial P, \tag{1c}$$

$$\mathbf{u}|_{t=0} = \mathbf{u_0}(\mathbf{x}) \quad \text{in } \Omega_f, \tag{1d}$$

where ρ_f is the fluid density, $\mathbf{u}$ the fluid velocity, and $\boldsymbol{\sigma}$ the fluid stress. $\mathbf{U}$ is ith particles translational velocity, and $\boldsymbol{\omega}$ is ith particles angular velocity. The initial velocity $\mathbf{u_0}$ satisfies the continuity equation. Only the Newtonian fluid is considered in this study, thus $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{D}$, with p being the fluid dynamic pressure, μ the dynamic viscosity and $\mathbf{D}$ the rate-of-strain tensor.

The governing equations in the particle domain are (we ignore the collision force for the time being)

$$m\frac{d\mathbf{U}}{dt} = \mathbf{F}^{\mathrm{H}} + \left(1 - \frac{1}{\rho_r}\right) m\mathbf{g}, \tag{2a}$$

$$\frac{d(\mathbf{J}\cdot\boldsymbol{\omega})}{dt} = \mathbf{T}^{\mathrm{H}}, \tag{2b}$$

where m, $\mathbf{J}$, $\mathbf{U}$ and $\boldsymbol{\omega}$ are the particle mass, moment of inertia tensor, translational velocity and angular velocity, respectively. $\mathbf{g}$ is the gravitational acceleration and ρ_r is the solid–fluid density ratio. $\mathbf{F}^{\mathrm{H}}$ and $\mathbf{T}^{\mathrm{H}}$ are the hydrodynamic force and torque on the particle, respectively, defined by

$$\mathbf{F}^{H} = \int_{\partial P} \mathbf{n}\cdot\boldsymbol{\sigma} ds, \tag{3a}$$

$$\mathbf{T}^{\mathrm{H}} = \int_{\partial P} \mathbf{r}\times(\mathbf{n}\cdot\boldsymbol{\sigma}) ds, \tag{3b}$$

where $\mathbf{n}$ is the unit outward normal on the particle surface and $\mathbf{r}$ is the position vector with respect to the particle mass center. $\boldsymbol{\sigma}$ is the fluid stress without the gravity. Note that the gravity term is absent in (1), which has no effect on the flow except producing a hydrostatic pressure and thereby a buoyance force on the particle. Since the buoyance force is not considered in $\mathbf{F}^{H}$, but it has effect on the total force rather than on the torque, we need to include it directly in the first equation of (2).

As in the DLM/FD method, the interior of the particle is filled with the fluid and a pseudo body-force is introduced over the particle inner domain to enforce the fictitious fluid to satisfy the rigid-body motion constraint, namely, the following equations are introduced for the interior of the particle:

$$\rho_f \frac{d\mathbf{u}}{dt} = \nabla\cdot\boldsymbol{\sigma} + \boldsymbol{\lambda} \quad \text{in } P, \tag{4a}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } P. \tag{4b}$$

Manipulating (2a) – $\int_P$ (4a) dx, (2b) – $\int_P \mathbf{r}\times$ (4a) dx, $\mathbf{F}^{H}$ and $\boldsymbol{\sigma}$ terms are eliminated to get the following the equations.

$$\left(1 - \frac{1}{\rho_r}\right) m\left(\frac{d\mathbf{U}}{dt} - \mathbf{g}\right) = -\int_P \boldsymbol{\lambda} dx \quad \text{in } P, \tag{5a}$$

$$\left(1 - \frac{1}{\rho_r}\right) \frac{(d\mathbf{J}\cdot\boldsymbol{\omega})}{dt} = -\int_P \mathbf{r}\times\boldsymbol{\lambda} dx \quad \text{in } P. \tag{5b}$$

Then the governing equations are non-dimensionalized by introducing the following scales: L_c for length, U_c for velocity, L_c/U_c for time, $\rho_f U_c^2$ for pressure p and $\rho_f U_c^2/L_c$ for the pseudo body-force. For convenience, we write the dimensionless quantities in the same form as their dimensional counterparts, unless otherwise specified. The dimensionless equations for the incompressible fluid can be written as follows:

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = \frac{\nabla\cdot\boldsymbol{\tau}}{Re} - \nabla p + \boldsymbol{\lambda} \quad \text{in } \Omega, \tag{6a}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega, \tag{6b}$$

$$\mathbf{u} = \mathbf{U} + \boldsymbol{\omega}\times\boldsymbol{r} \quad \text{in } P, \tag{6c}$$

$$(\rho_r - 1)V_p^*\left(\frac{d\mathbf{U}}{dt} - Fr\frac{\mathbf{g}}{g}\right) = -\int_P \boldsymbol{\lambda} dx \quad \text{for } P, \tag{6d}$$

$$(\rho_r - 1)\frac{(d\mathbf{J}^*\cdot\boldsymbol{\omega})}{dt} = \int_P \mathbf{r}\times\boldsymbol{\lambda} dx \quad \text{for } P. \tag{6e}$$

In the above equations, Re represents the Reynolds number defined by $Re = \rho_f U_c L_c/\mu$, Fr the Froude number defined by $Fr = \frac{gL_c}{U_c^2}$, V_p^* the dimensionless particle volume define by $V_p^* = \frac{m}{\rho_s L_c^d}$ and $\mathbf{J}^*$ the dimensionless moment of inertia tensor defined by $\mathbf{J}^* = \frac{\mathbf{J}}{\rho_s L_c^{d+2}}$, where ρ_s is the particle density and d is the dimensionality of the problem involved. Note that the pseudo body-force $\boldsymbol{\lambda}$ in (6) is defined only in the solid domain P.

2.2. Numerical implementation

A finite-volume method using a staggered grid for incompressible flow is implemented. The SIMPLE algorithm is used to solve the fluid equations with modifications to account for the presence of particles. The implicit Euler scheme is used for time discretization.

The discretised momentum equations for 2-D flows in the x and y directions

$$a_{i,J}u^*_{i,J} = \sum a_{nb}u^*_{nb} - \frac{p^*_{I,J} - p^*_{I-1,J}}{\delta x}\Delta V_u + b_{i,J} + \lambda^*_{xi,J}, \tag{7}$$

$$a_{I,j}v^*_{I,j} = \sum a_{nb}v^*_{nb} - \frac{p^*_{I,J} - p^*_{I-1,J}}{\delta y}\Delta V_v + b_{I,j} + \lambda^*_{yI,j}. \tag{8}$$

The pressure correction equation is

$$a_{I,J}p'_{I,J} = a_{I-1,J}p'_{I-1,J} + a_{I+1,J}p'_{I+1,J} + a_{I,J-1}p'_{I,J-1} + a_{I,J+1}p'_{I,J+1} + b'_{I,J}. \tag{9}$$

After solving Eqs. (7)–(9), correct the pressure and velocity as traditional SIMPLE do

$$p_{I,J} = p^*_{I,J} + p'_{I,J}, \tag{10}$$

$$u_{i,J} = u^*_{i,J} + d_{i,J}(p'_{I-1,J} - p'_{I,J}), \tag{11}$$

$$v_{I,j} = v^*_{I,j} + d_{I,j}(p'_{I,J-1} - p'_{I,J}), \tag{12}$$

then, calculate $\mathbf{U}^{n+1}$, $\boldsymbol{\omega}^{n+1}$ according to the (13) and (14),

$$\rho_r V^*_p \frac{\mathbf{U}^{n+1}}{\Delta t} = (\rho_r - 1)V^*_p\left(\frac{\mathbf{U}^n}{\Delta t} + Fr\frac{\mathbf{g}}{g}\right) + \int_P \left(\frac{\mathbf{u}}{\Delta t} - \lambda^*\right)d\mathbf{x}, \tag{13}$$

$$\rho_r \frac{(\mathbf{J}^*\cdot\boldsymbol{\omega}^{n+1})}{\Delta t} = (\rho_r - 1)\left[\frac{\mathbf{J}^*\cdot\boldsymbol{\omega}^n}{\Delta t} - \boldsymbol{\omega}^n\times(\mathbf{J}^*\cdot\boldsymbol{\omega}^n)\right] + \int_P \mathbf{r}\times\left(\frac{\mathbf{u}}{\Delta t} - \boldsymbol{\lambda}^*\right)d\mathbf{x}, \tag{14}$$

and update the body force:

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^* + \frac{\mathbf{U}^{n+1} + \boldsymbol{\omega}^{n+1}\times\mathbf{r} - \mathrm{I}(\mathbf{u})}{\Delta t}, \tag{15}$$

where I is interpolation operator that transfers fluid velocity from Eulerian nodes to Lagragian nodes.

Transfer the force correction $(\boldsymbol{\lambda} - \boldsymbol{\lambda}^*)$ to Eulerian nodes to update fluid velocity

$$\mathbf{u}^{n+1*} = \mathbf{u} + \Delta t D(\boldsymbol{\lambda} - \boldsymbol{\lambda}^*), \tag{16}$$

where the D is the distribution operator that transfers particle velocity from Lagragian nodes to Eulerian nodes.

Then set $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}$, $\mathbf{u} = \mathbf{u}^{n+1*}$, $p^* = p$, loop Eqs. (7)–(15), until the equations are convergence. After convergence, we go to the next time level and update the position of particles

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \Delta t(\mathbf{U}^{n+1} + \boldsymbol{\omega}^{n+1}\times\mathbf{r}^{n+1} + \mathbf{U}^n + \boldsymbol{\omega}^n\times\mathbf{r}^n)/2. \tag{17}$$

When the position of particles is changed, the collision should be checked. If collision happens, DEM is used to update the position.

Because $\boldsymbol{\lambda}$ is defined at the Lagrangian nodes instead of (13), (14) and (16), which will be discussed later, it needs to be transferred to Eulerian nodes. Also $\mathbf{u}$ need to be interpolated to Lagrangian nodes for imposing rigid body motion. Different nodes are shown in Fig. 1.

Therefore, we use the discrete δ-function to transfer a quantity between the Eulerian and lagrangian frames explicitly as in the immersed boundary method. This can improve the computational efficiency than the Uzawa iteration [11,12]. In the present study, we adopt the linear function (bi-linear for 2D) as a discrete approximation to the δ-function. For the case of 3D, the discrete δ-function is defined by

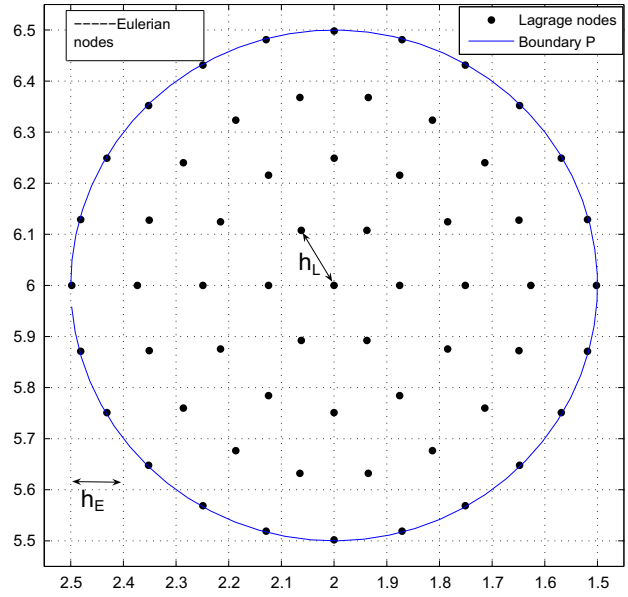

Fig. 1. Arrangements of Lagrangian nodes in case of a circular particle in the Eulerian nodes.

$$\delta_h(\mathbf{r}) = d_h(\mathbf{r}_x)\cdot d_h(\mathbf{r}_y)\cdot d_h(\mathbf{r}_z),$$

where r_x, r_y and r_z denote the components of $\mathbf{r}$, and $d_h(r)$ is define by

$$d_h(r) = \begin{cases} 1 & -\left|\frac{r}{h}\right| \text{ for } |r| < h, \\ 0 & \text{otherwise}, \end{cases}$$

in which h is the mesh size of the homogeneous Eulerian grid. Using the discrete δ-function, a discrete quantity f is transferred between the Eulerian (x_i) and Lagrangian (X_I) frames as follows:

$$I = f_{E\to L} : f_L(X_I) = \sum_i f_E(x_i)\delta_h(x_i - X_I),$$

$$D = f_{L\to E} : f_E(x_i) = \sum_I f_L(X_I)\delta_h(x_i - X_I)\Delta V^*_I,$$

where $f_E(x_i)$ and $f_L(X_I)$ represent the values of f at the Eulerian nodes x_i and at the Lagrangian nodes X_I, respectively, ΔV^*_I is the ratio of length size of the control volume for Lagrangian node to that for Eulerian node.

When the collision happens, the DEM is used in place of (13) and (14). It means that a new force F_{DEM} is added to the particles:

$$\begin{cases} \mathbf{F}_{DEM}(t) + m\mathbf{g} - \beta\mathbf{U} = m\frac{d\mathbf{U}}{dt}, \\ \mathbf{M}(t) - \beta\boldsymbol{\omega} = \mathbf{J}\frac{d\boldsymbol{\omega}}{dt}, \end{cases} \tag{18}$$

where $\mathbf{M}(t) = \mathbf{F}_{DEM}(t)\times\mathbf{r}$ is the torque, and β is damping coefficient for $\beta = 0$ means that there is no damping. In this paper, we adopt Hertz–Mindlin (no slip) model to deal with the force:

$$\mathbf{F}_{DEM} = k\alpha + c\dot{\boldsymbol{\alpha}},$$

where k is the linear spring stiffness, c is the dashpot coefficient, $\boldsymbol{\alpha}$ is the overlap, and $\dot{\boldsymbol{\alpha}}$ is the overlap velocity. Because the DEM just works for short time, we use the mixed forward Euler-trapezoidal rule method to solve Eq. (18), just as follows

$$\mathbf{U}(t) = \mathbf{U}(t - \Delta t_i)\frac{m/\Delta t_i - \beta/2}{m/\Delta t_i + \beta/2} + \frac{\mathbf{F}_{DEM} + m\mathbf{g}}{m/\Delta t_i + \beta/2},$$

$$\boldsymbol{\omega}(t) = \boldsymbol{\omega}(t - \Delta t_i)\frac{J/\Delta t_i - \beta/2}{J/\Delta t_i + \beta/2} + \frac{\mathbf{M}(t)}{J/\Delta t_i + \beta/2},$$

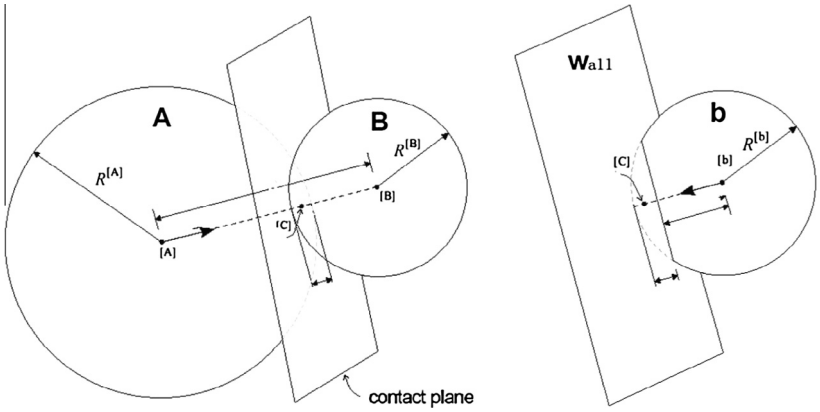

Fig. 2. The contact model of DEM (left) two particle particles collide; (right) the particle collides with the wall.

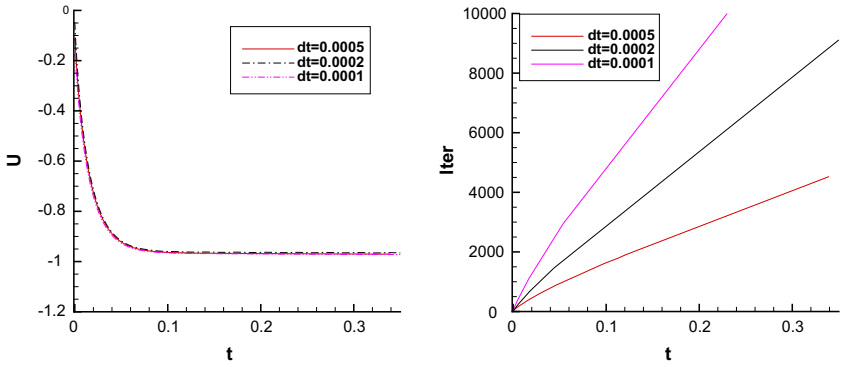

Fig. 3. Time development of the settling velocity of a circular particle in a vertical channel for (left) U of the particle and (right) number of SIMPLE methods accumulative total iterations under different time steps. Where $Re = 0.1,\ W/D = 4,\ h = D/16,\ \rho_r = 1.2,\ N_a = 6$.

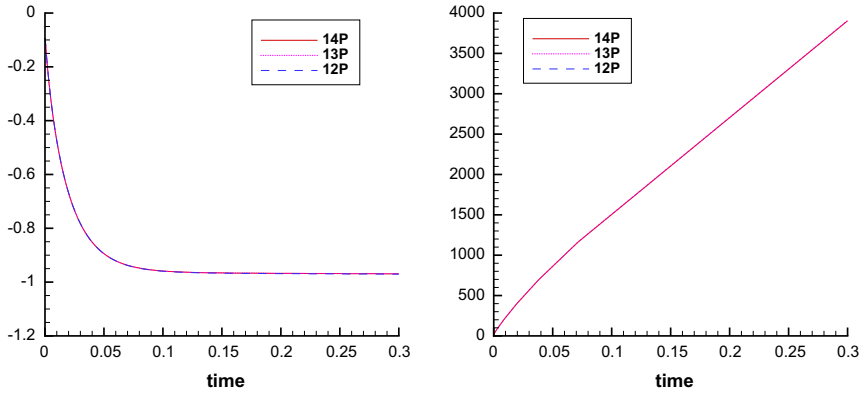

Fig. 4. Time development of the settling velocity of a circular particle in a vertical channel (left) U of the particle (right) number of SIMPLE methods total iterations, with different numbers of ring for different time-step. Where $Re = 0.1,\ W/D = 4,\ \rho_r = 1.2,\ h = D/32,\ N_a = 12, 13, 14$.

where $\Delta t_i = \Delta t/N_i$ is the DEMs time step according to particles material. The contact model of DEM is showed in Fig. 2.

3. Numerical experiments

3.1. Sedimentation of a circular particle in channel

Throughout the present study, the results are computed and presented in the dimensionless form, thus we need to define the characteristic velocity and length for each case. For low Reynolds numbers, there exists an analytical expression for the drag force on a circular particle settling in a channel at the velocity U:

$$F_d = 4\pi K \mu U, \tag{19}$$

where K is a constant related to the effect of the channel width on the drag force and can be expressed in terms of the ratio of the channel width to the particle diameter W^* (i.e., W/D) [8]:

$$K = \frac{1}{\ln W^* - 0.9157 + 1.7244/(W^*)^2 - 1.7302/(W^*)^4 + 2.4056/(W^*)^6 - 4.5913/(W^*)^8}. \tag{20}$$

By taking the characteristic velocity as U_c

$$U_c = \frac{D^2}{16K\mu}(\rho_s - \rho_f)g, \tag{21}$$

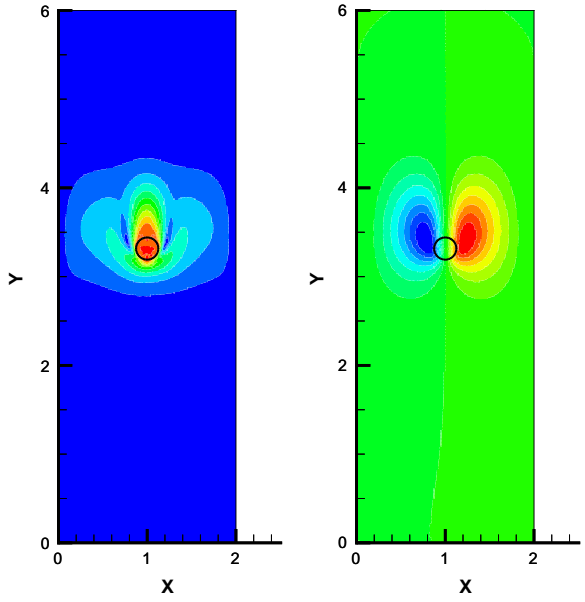

Fig. 5. Particle position and flow field visualization: velocity contour (left) and streamline (right) at $t = 0.4(\rho_s = 1.25, \mu = 0.1, h_E = 1/160, \Delta t = 0.005)$.

the dimensionless terminal settling velocity is expected to be unity. The characteristic length is the particle diameter. We take the Reynolds number Re and the density ratio ρ_r as the independent dimensionless control parameters, and then the Froude number Fr is not independent, but can be expressed in terms of Re and ρ_r as follows:

$$Fr = \frac{16K}{(\rho_r - 1)Re}. \tag{22}$$

We will consider the sedimentation of a circular particle in a vertical channel at $Re = 0.1$. The characteristic velocity is defined by (21) for $Re = 0.1$ and is expected to be unity. For the circular particle, we distribute the collocation points on the concentric rings as show in Fig. 1: one point at the particle center, and $6i$ points on the ith ring for $i = 1, \ldots, N_a$, where N_a is the number of the rings. Each Lagrangian point's distance is h_L. In order to avoid the possibility that there are two collocation nodes in a Cartesian cell, $h_L \geqslant 1.45h$ is good choice. For $h = D/16$ (D is circular particle's diameter), $N_a = 6$ is good choice. The particle is released at the center of the channel. Fig. 3 shows the time development of the settling velocity at different Δt for $Re = 0.1$, $W/D = 4$, $\rho_r = 1.2$, and Eulerian mesh size $h = D/16$. From Fig. 3 (left), we can see that the results. The results exhibit a satisfactory time-step independence. Using $\Delta t = 0.0005$, the simulation of the acceleration process from zero velocity to the steady-state velocity (almost to the unity) only requires less than 200 time steps. The characteristic velocity is well within the range of results that theoretical analysis gets (21). At the same time, the total iterations developments of SIMPLE are show in Fig. 3 (right). These show that the total number of iteration changes linearly with the time. This is because of the flow field changes little for the small $Re = 0.1$. This experiment shows our code is right.

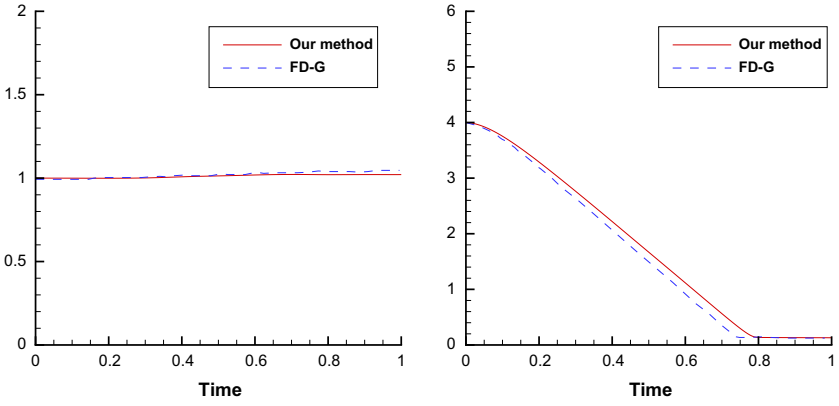

Fig. 6. Histories of the x-coordinate (left) and y-coordinate (right) of the center of the disk for $\rho_s = 1.25$ and $\mu = 0.1$.

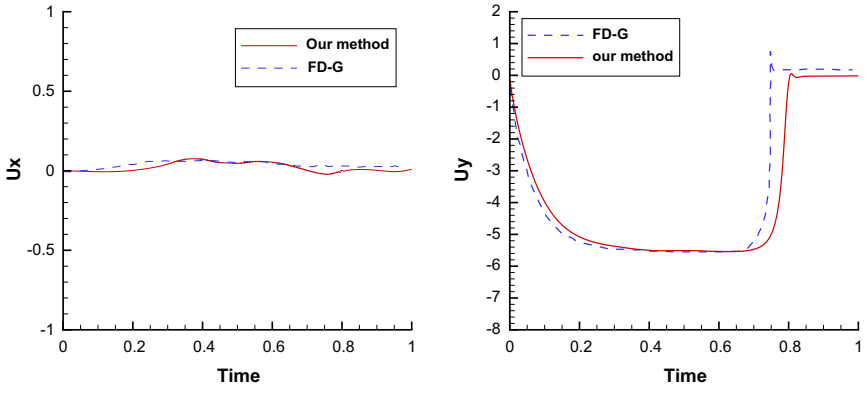

Fig. 7. Histories of the x-component (left) and y-component (right) of translation velocity of the disk for $\rho_s = 1.25$ and $\mu = 0.1$.

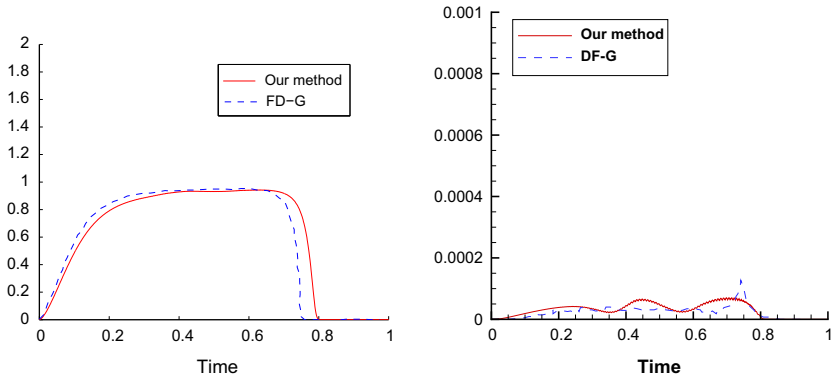

Fig. 8. Histories of the translational (left) and rotational (right) kinetic energies of the disk for $\rho_s = 1.25$ and $\mu = 0.1$.

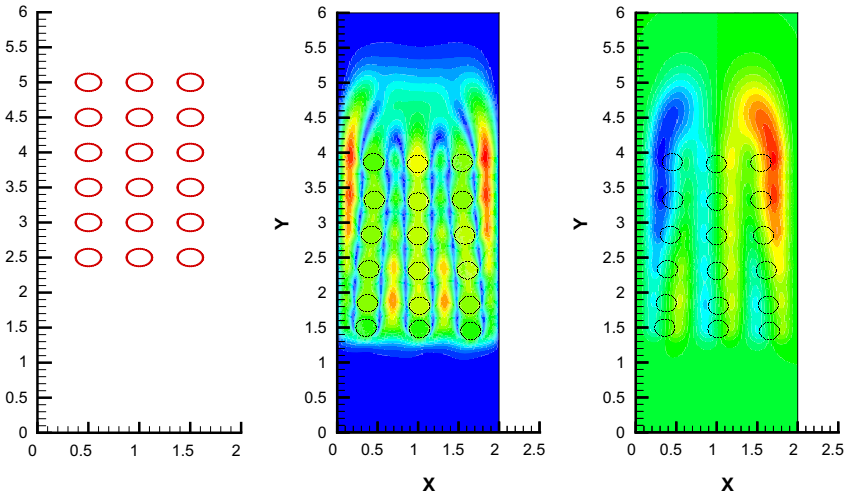

Fig. 9. Sedimentation of 18 disks in a closed box starting from an initial top-packed (left) layout: particles pattern with vertical velocity contours (middle) and (right) streamlines contours at $t = 0.26$.

We also performed numerical test by varying h_L for $Re = 0.1$. Fig. 4 shows the time development of the settling velocity and SIMPLE methods total iterations for different numbers of rings of Lagrangian nodes: $N_a = 12, 13, 14$. We can see that for different N_a, results are almost the same. The steady-state velocity of particle is almost unity. The total iterations' development of SIMPLE is not change. From this, it is known that when h_L changes in the vicinity of $1.45h$, the code's cost time is almost the same.

For larger Re, we compute the same Example 8.3.3 in Ref. [12]. We set initial computational domain is $\Omega = (0, 2) \times (0, 6)$ and the fluid velocity is 0. The boundary of Ω is wall. The fall of the disk in the viscous fluid has been simulated for $d = 0.25$, $\rho_r = \frac{\rho_{solid}}{\rho_{fluid}} = 1.25$ and $\mu = 0.1$. We set $Re = \rho_f U L_c / \mu \approx 30$. Assuming that at $t = 0$ the center G of disk is located at $(1, 4)$. The Eulerian mesh size $h_E = 1/64$ and the Lagrangian size is $h_L = D/14$. The time step $\Delta t = 0.005$. The DEM parameters used in the collision model are $\beta = 0$, $k = 150$ and $c = 0$. Fig. 5 shows the flow and the particle position at $t = 0.4$. Other results and comparisons with Example 8.3.3 of Ref. [12] are shown in Figs. 6–8.

Fig. 7 (right) shows that the cylinder quickly reaches a uniform falling velocity until it hits the bottom of the cavity. A careful examination of Fig. 6 shows that a symmetry breaking of small amplitude is taking place with the disk moving slightly on the right, away from the vertical symmetry axis of the cavity. Fig. 8 shows that the rotational component of the kinetic energy is small compared with the translational component. The differences between present and those in Ref. [12] are small. From Fig. 7 (left), we can see that x-component of translation velocity of the disk is smaller than in Ref. [12]. Therefore the disk of present calculation is less deviated to the right shown in Fig. 6 (left). This shows that our method is more closer to the reality. In Fig. 7 (right), we can see that y-component of translation velocity of the disk changed more slowly than Ref. [12] at the beginning. However, steady state falling velocity is the almost the same. In Ref. [12], artificial repulsive force was used to make the particle do not reach the wall. Our collision use the DEM that allows the particle to contact with the wall and bounce back elastically. Therefore, the y-component of translation velocity of our method is more slowly approaching to zero.

The translational kinetic energy of the disk in our method is smaller than that in Ref. [12] at first and approaches to zero more slowly. This is consistent with the velocity of the particle in Fig. 7. From Figs. 7 and 8 (right), when the particle contacts with the wall at $t = 0.75$, the velocity and rotational kinetic energies of disk in Ref. [12] have a big change because of the artificial force added to particle. However, our method is more smooth and more in agreement with practice.

3.2. Sedimentation of several circular particles in channel

We now consider the sedimentation of 18 circular particles in a vertical channel. We set the same computational parameters as the

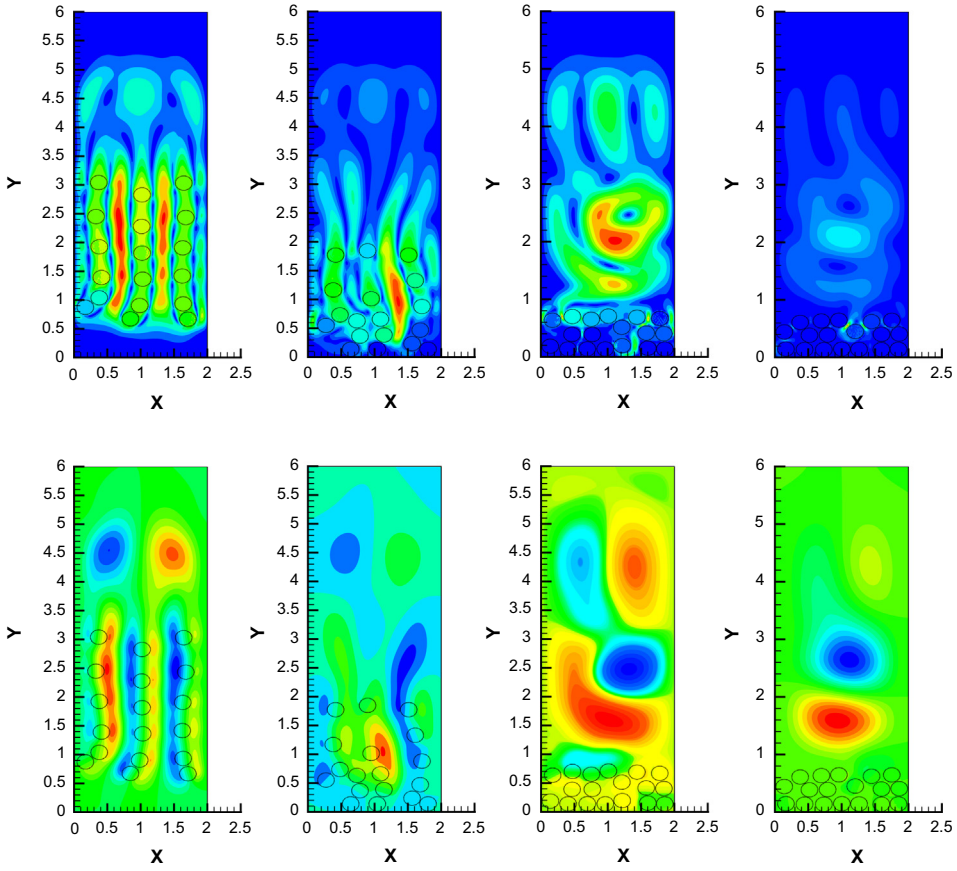

Fig. 10. Sedimentation of 18 disks in a closed box (Up) vertical velocity contours and (Down) streamlines contours at $t = 0.4,\ t = 0.58,\ t = 1,\ t = 1.8$ from left to right.

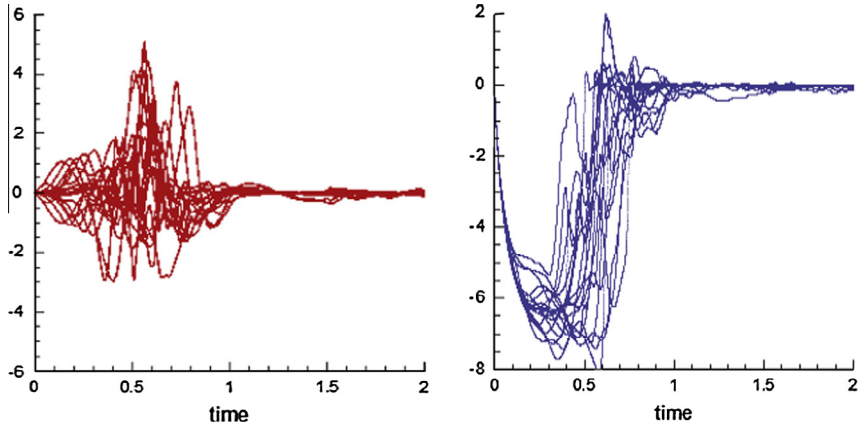

Fig. 11. Histories of the x-component (left) and y-component (right) of translation velocity of the 18 particles.

second example in this paper. Assuming that at $t = 0$, the 18 particles are located at $G_{ij}(0) = (x_i, y_j),\ x_i = 0.5 + 2(j-1),\ y_j = 6 - 2(i-1),\ i = 1, \ldots, 6,\ j = 1, 2, 3$. Figs. 9 and 10 present the particles patterns together with vertical velocity and streamlines contours.

We see that the flow regime is clearly regular at the beginning time. However, particles have collisions after some time and some particles temporarily move up. The DEM solver handles collisions between particle–particle and particle–wall without any trouble. Simulation ends as all particles have settled to the bottom of the box.

For the more detail, we can see the histories of the x-component (left) and y-component (right) of translation velocity of the disks from Figs. 11–13. When $t \in (0, 0.1)$, the particles move just like one particle. But as time goes on the upper particle will sick the lower particles until the collision happens. When $t \in (0.1, 0.3)$, the collision did not happen. The middle-column 6 particles almost remain in the center. However, some of left-column 6 particles will move left and some of right 6 particles move right. When $t \in (0.3, 0.5)$, some particles have collisions with other particles, but the collision of particles-wall does not happen. When $t \in (0.5, 1)$, the particle–particle collision and particle–wall collision happen. This makes the particle settle to the bottom of the box. When $t \geqslant 1$, the particles will pack together and the velocity trends to zero. From Fig. 13, we can see that the number of SIMPLE methods total iterations increase linearly with the time. However, when the collision happens for $t \in (0.3, 1)$, the number of iterations

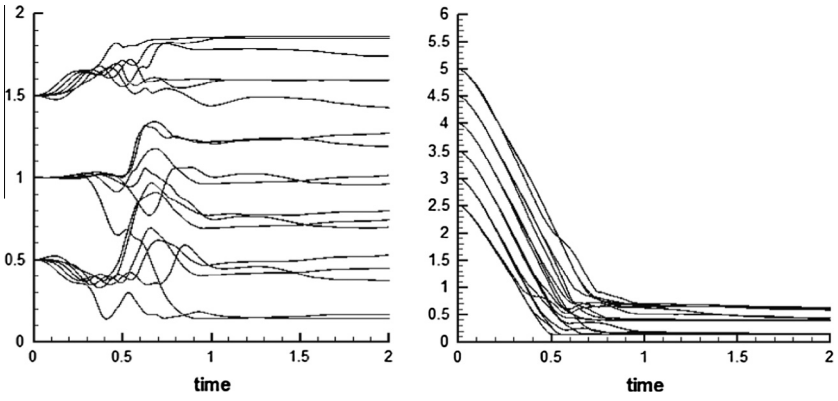

Fig. 12. Histories of the x-coordinate (left) and y-coordinate (right) of the center of the 18 particles.

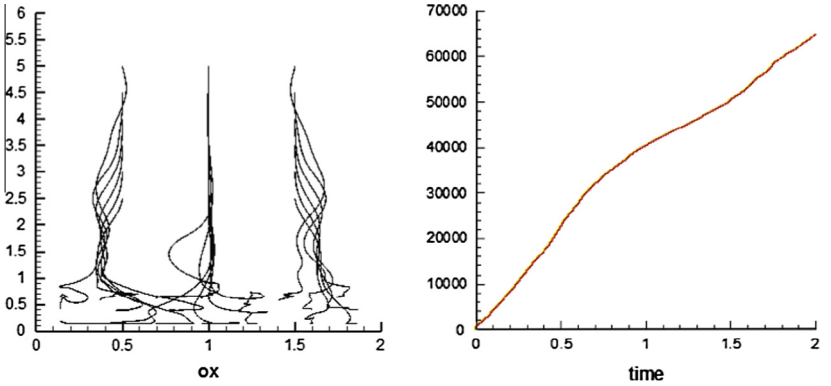

Fig. 13. Histories of the trajectory (left) of the center of the 18 particles, (right) number of SIMPLE methods total iterations.

will also non-linear increase. The more frequent the collision, the more the number of iterations.

4. Conclusion

A hybrid FD-DEM solver is developed for rigid particulate flow with collision. A collision strategy based on DEM is introduced for multi-particle system. In this method, the contact force in particle–particle and particle–wall is calculated and the method can be generalized for collisions including friction. Several benchmark examples are given to show the effectiveness of the present method.

Acknowledgements

The authors thank Dr. Wei Liu for useful discussions and valuable comments and thank the anonymous referees for helpful suggestions. This work was supported by the Natural Science Foundation of China (10972230, 11021101, 11261160486), state key program for developing basic sciences (2010CB731505) and Department of Education of Guangdong Province (2014KQNCX175).

References

[1] Davis RH. Elastohydrodynamic collisions of particles. PCH Physico Chem Hydrodyn 1987;9:41–52.
[2] Mittal R, Iaccarino G. Immersed boundary methods. Annu Rev Fluid Mech 2005;37:239–61.
[3] Marella S, Krishnan S, Liu H, Udaykumar HS. Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations. J Comput Phys 2005;210:1–31.
[4] Glowinski R, Pan T-W, Pe'riaux J. A fictitious domain method for Dirichlet problems and applications. Comput Meth Appl Mech Eng 1994;111:283–303.
[5] Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J Comput Phys 2000;161:5–60.
[6] Le DV, Khoo BC, Peraire J. An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. J Comput Phys 2006;220:109–38.
[7] Xu S, Wang ZJ. An immersed interface method for simulating the interaction of a fluid with moving boundaries. J Comput Phys 2006;216:454–93.
[8] Happel J, Brenner H. Low Reynolds number hydrodynamics. New York: Prentice-Hall; 1965.
[9] Yu ZS, Shao XM. A direct-forcing fictitious domain method for particulate flows. J Comput Phys 2007;227:292–314.
[10] Ardekani AM, Dabiri S, Rangel RH. Collision of multi-particle and general shape objects in a viscous fluid. J Comput Phys 2008;227:10094–107.
[11] Glowinski R, Pan T-W, Hesla TI, Joseph DD. A distributed Lagrange multiplier/fictitious domain method for particulate flows. Int J Multiphase Flow 1999;25:755–94.
[12] Glowinski R, Pan T-W, Hesla TI, Joseph DD, Periaux J. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. J Comput Phys 2001;169:363–426.
[13] Ye Y, Li KL. Entropic Lattice Boltzmann Method based high Reynolds number flow simulation using CUDA on GPU. Comput Fluids 2013;88:241–9.
[14] Peskin CS. Numerical analysis of blood flow in the heart. J Comput Phys 1977;25:220–52.