INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **51**:531–566
Published online 20 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1129

Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method

Decheng Wan[‡] and Stefan Turek[*,†]

Institute of Applied Mathematics LS III, University of Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany

SUMMARY

Direct numerical simulation techniques for particulate flow by the fictitious boundary method (FBM) are presented. The flow is computed by a multigrid finite element solver and the solid particles are allowed to move freely through the computational mesh which can be chosen independently from the particles of arbitrary shape, size and number. The interaction between the fluid and the particles is taken into account by the FBM in which an explicit volume based calculation for the hydrodynamic forces is integrated. A new collision model based on papers by Glowinski, Joseph, Singh and coauthors is examined to handle particle–particle and particle–wall interactions. Numerical tests show that the present method provides a very efficient approach to directly simulate particulate flows with many particles. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: particulate flows; incompressible Navier–Stokes equations; multigrid; FEM; fictitious boundary method

1. INTRODUCTION

Liquids containing large solid particles are common in many industrial processes, such as foods containing particles, slurry flows, mining extraction, fluidization of catalyst beds, separation process using cyclones, etc. The phenomena of such particulate flows are also visible everywhere around our living environments, for instance flow around high-rise buildings, the drag force induced by driving a car accelerating in the wind, ocean current interaction with the offshore structures, sedimentation flow in estuary and sand flow in desert, etc. From the numerical point of view, particulate flows are quite hard to simulate since both the incompressible fluid velocity and the domain in which it is defined are unknown. It can require a

*Correspondence to: Stefan Turek, Institute of Applied Mathematics LS III, University of Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany.
†E-mail: Stefan.Turek@math.uni-dortmund.de
‡E-mail: Wan.Decheng@math.uni-dortmund.de

Received 27 April 2005
Revised 26 September 2005
Accepted 26 September 2005

huge amount of time for the frequent generation or deformation of the computational grids when the corresponding boundaries are complex and changing in time; especially for the case with large numbers of particles (greater than 10 000), the interaction between fluid and particles as well as the collision between particles gives further complexity to the problem.

So far, such problems have motivated the development of numerous algorithms which can be broadly classified into two families. The first one is an Eulerian approach which uses a fixed mesh (or a mesh independent of the particles) that covers the whole domain where the fluid may be present. One popular example of this approach is the class of distributed Lagrange multiplier (DLM)/fictitious domain methods proposed by Glowinski, Joseph and coauthors [1–3]. The second type is a Lagrangian approach which is based on a moving mesh following the motion of the boundary of the particles in the fluid. As the mesh motion within the fluid can be defined arbitrarily, this approach is usually referred to as Arbitrary Lagrangian Eulerian (ALE). Hu, Joseph and Crochet [4, 5] as well as Maury [6, 7] have applied the ALE to fluid–particle flows. One big advantage of the Eulerian approaches over the Lagrangian approaches is that the computational mesh can remain unchanged such that the CPU cost per time step can be significantly decreased—less computational effort due to saving the expensive mesh generation—and special techniques can be easily incorporated into standard CFD codes which mostly allow fixed computational grids without local adaptivity; however, the resulting accuracy is often not clear. Therefore, our overall aim is to deal successfully with moving boundaries such that the accuracy of the numerical approximation is sufficiently high while at the same time also the computational cost is significantly decreased.

In the spirit of the Eulerian approaches, we present an efficient fictitious boundary method (FBM) for the detailed simulation of particulate flows. The method is based on an unstructured FEM background grid. The motion of the solid particles is modelled by the Newton–Euler equations. Based on the boundary conditions applied at the interface between the particles and the fluid which can be seen as an additional constraint to the governing Navier–Stokes equations, the fluid domain can be extended into the whole domain which covers both fluid and particle domains. The FBM starts with a coarse mesh which may contain already many of the geometrical fine-scale details, and employs a (rough) boundary parametrization which sufficiently describes all large-scale structures with regard to the boundary conditions. Then, all fine-scale features are treated as interior objects such that the corresponding components in all matrices and vectors are unknown degrees of freedom which are implicitly incorporated into all iterative solution steps. The considerable advantage of the FBM is that the total computational domain does not have to change in time, and has to be meshed only once—or more precisely: it can be handled independently from the flow features [8, 9].

For studying the interaction between fluid and solid, the efficient and accurate calculation of forces acting on the moving rigid bodies is very important. However, in the FBM, it is not straightforward to compute explicitly these forces, because the hydrodynamic drag and lift forces acting on the moving solid bodies are a very delicate quantity. They include the results directly on the wall surface of the moving rigid bodies which however is represented only implicitly in the fictitious boundary method due to the use of a fixed grid rather than a body-conformal grid. The FBM differs only between ‘inside’ and ‘outside’ of a particle, but it does not define the wall surface of the particle. Therefore, the integration of forces acting only on the wall surface of the rigid bodies cannot be implemented directly in the fictitious boundary method. For overcoming this difficulty, a volume integration instead of the conventional surface integral for the calculation of the hydrodynamic forces by introducing an

auxiliary function has been proposed in References [10, 11]. Obviously, in such volume based calculations, the explicit reconstruction of the wall surface of the moving rigid bodies can be avoided. In this paper, we expand the implementation of this volume integral approach with an auxiliary function to the finite element method and the FBM, and we introduce corresponding data structures for the efficient handling of many particles.

For treating more than one particle in fluid, a collision model is needed to prevent particles from interpenetrating each other. A systematic study of collisions of particles in the flow can be found in Reference [12]. Collisions or near-collisions between the particles present severe difficulties in direct simulations of particulate flows. Even near-collisions can significantly increase the cost of a simulation, because in order to simulate the particle–particle interaction mechanism in a direct manner, the flow fields in the narrow gap between the converging particle surfaces must be accurately resolved. The corresponding element size in Lagrangian approaches has to be decreased, leading to extremely small elements and significantly increased numbers of unknowns to be solved for. In such simulations, numerical problems are likely to occur when the particles get very close to each other, i.e. the mesh is to be refined in the gap zone, therefore it is computationally expensive. To handle this problem numerically, several collision models have been proposed in the literature. Glowinski, Joseph, Pan and coauthors [1–3] have described repulsive force models to prevent the particles from overlapping each other. Singh *et al.* [13] proposed another repulsive force model which allows the particles to come arbitrarily close and even slightly to overlap each other. In addition to using the concept of repulsive forces to construct collision models, there are also other ways to form such models, for example, conservation collision models which are based on the conservation of linear momentum and kinetic energy [14], lubrication collision models [15] and stochastic collision models [16] which are based on physical properties of the particles, as well as semi-experiential collision models [17], etc. In this paper, following those models proposed by Glowinski, Joseph, Singh and coauthors, we describe a new repulsive force model which cannot only prevent the particles from getting too close to each other, it can also deal with the case of particle overlapping when numerical simulations bring the particles very close or even overlapping due to unavoidable numerical truncation errors.

The subsequent parts of the paper are arranged as follows: In Section 2, the governing equations for the coupled system of fluid and particles are given, and a new treatment of collisions based on those models proposed by Glowinski, Joseph and Singh is presented. The numerical solution process for solving the coupled system of fluid and particles is described in Section 3. The explicit fictitious boundary method (FBM) and the proposed volume based integration for hydrodynamic force calculations are presented. Then, the numerical discretization, solution procedure, the algorithmic details, and the time reducing techniques for dealing with large numbers of particles are given. In Section 4, extensive numerical test problems are presented. First of all, two configurations of two-dimensional flow around a circular body in a channel are used for the validation of hydrodynamic force calculations based on the volume integration in context of the presented multigrid FBM. Then, flows with one rotating and moving particle are examined to validate the prediction of angular and translational velocities by the presented FBM. Finally, several cases of multiple particles in a fluid are examined to evaluate the collision model and the efficiency of the presented FBM for the simulation of particulate flows with large numbers of particles. The paper ends with a conclusion in Section 5.

2. GOVERNING EQUATIONS

2.1. Model of incompressible flow

Our aim is to compute incompressible flow in interaction with N particles of mass M_i ($i=1, \ldots, N$) in a fluid with density ρ_f and viscosity v. Denote $\Omega_f(t)$ as the domain occupied by the fluid at time t, and $\Omega_i(t)$ as the domain occupied by the ith particle. So, the fluid flow obeys the following Navier–Stokes equations in $\Omega_f(t)$

$$\rho_f \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \sigma = 0, \quad \nabla \cdot \mathbf{u} = 0 \quad \forall t \in (0, T) \tag{1}$$

where σ is the total stress tensor in the fluid phase, defined as

$$\sigma = -p\mathbf{I} + \mu_f [\nabla \mathbf{u} + (\nabla \mathbf{u})^T] \tag{2}$$

Here, $\mathbf{I}$ is the identity tensor, $\mu_f = \rho_f \cdot v$, p is the pressure and $\mathbf{u}$ is the fluid velocity. Let $\Omega_T = \Omega_f(t) \cup \{\Omega_i(t)\}_{i=1}^N$ be the entire computational domain which shall be independent of t. Dirichlet- and Neumann-type boundary conditions can be imposed on the outer boundary $\Gamma = \partial\Omega_f(t)$. Since $\Omega_f = \Omega_f(t)$ and $\Omega_i = \Omega_i(t)$ are always depending on t, we drop t in all following notations.

2.2. Model of particle motion

In the fluid domain, the particles are allowed to translate and rotate with gravity, fluid forces acting on them and collision forces (repulsive forces or lubrication forces) in particle–particle or particle–wall interactions. The equations that govern this motion for each particle are the following Newton–Euler equations, i.e. the translational velocities $\mathbf{U}_i$ and angular velocities ω_i of the ith particle satisfy

$$M_i \frac{d\mathbf{U}_i}{dt} = (\Delta M_i)\mathbf{g} + \mathbf{F}_i + \mathbf{F}_i', \quad \mathbf{I}_i \frac{d\omega_i}{dt} + \omega_i \times (\mathbf{I}_i \omega_i) = T_i \tag{3}$$

where M_i is the mass of the ith particle ($i = 1, \ldots, N$), $\mathbf{I}_i$ is the moment of the inertia tensor, ΔM_i is the mass difference between the mass M_i and the mass of the fluid occupying the same volume, $\mathbf{g}$ is the gravity vector, $\mathbf{F}_i'$ are collision forces acting on the ith particle due to other particles which come close to each other. We assume that the particles are smooth without tangential forces of collisions acting on them. The details of the collision model will be discussed in the following subsection. $\mathbf{F}_i$ and T_i are the resultants of the hydrodynamic forces and the torque about the centre of mass acting on the ith particle which are calculated by

$$\mathbf{F}_i = (-1) \int_{\partial\Omega_i} \sigma \cdot \mathbf{n} \, d\Gamma_i, \quad T_i = (-1) \int_{\partial\Omega_i} (\mathbf{X} - \mathbf{X}_i) \times (\sigma \cdot \mathbf{n}) \, d\Gamma_i \tag{4}$$

where σ is the total stress tensor in the fluid phase defined by Equation (2), $\mathbf{X}_i$ is the position of the mass centre of the ith particle, $\partial\Omega_i$ is the boundary of the ith particle, $\mathbf{n}$ is the unit normal vector on the boundary $\partial\Omega_i$ pointing outward of the flow region. The position $\mathbf{X}_i$ of

the ith particle and its angle θ_i are obtained by integration of the kinematic equations, i.e.

$$\frac{\mathrm{d}\mathbf{X}_i}{\mathrm{d}t} = \mathbf{U}_i, \quad \frac{\mathrm{d}\theta_i}{\mathrm{d}t} = \omega_i \tag{5}$$

No-slip boundary conditions are applied at the interface $\partial\Omega_i$ between the ith particle and the fluid, i.e. for any $\mathbf{X} \in \bar{\Omega}_i$, the velocity $\mathbf{u}(\mathbf{X})$ is defined by

$$\mathbf{u}(\mathbf{X}) = \mathbf{U}_i + \omega_i \times (\mathbf{X} - \mathbf{X}_i) \tag{6}$$

2.3. Collision models

If there are two or more particles in the fluid, a collision model is needed to prevent the particles from interpenetrating each other. Theoretically, smooth particle–particle or boundary wall–particle collisions do not take place in finite time in the continuous system since there are repulsive forces (or lubrication forces) to prevent these collisions in the case of viscous fluids [2, 3, 18]. However, collisions occur in nature and particles can contact each other in numerical simulations since the gap can become exceedingly small if special precautions are not taken, and numerical errors may also allow contact or even overlap in simulations. Collisions between particles present severe difficulties in direct simulations of particulate flows. Even near-collisions can significantly increase the cost of a simulation, because in order to simulate the particle–particle interaction mechanism, the flow in the narrow gap between the converging particles must be accurately resolved. The required element size decreases with the gap width, leading to extremely small elements and significantly increased numbers of unknowns to be solved for.

For simplicity, in most cases it is assumed that the collisions are smooth, which means that if two rigid bodies collide under the effect of gravity and hydrodynamical forces, the rigid body velocities coincide at the points of contact. Glowinski, Joseph and coauthors [2, 3] proposed a repulsive force model in which an artificial short-range repulsive force between particles is introduced keeping the particle surfaces more than one element (the range of the repulsive force) apart from each other. In this model, overlapping of the regions occupied by the rigid bodies is not allowed since conflicting rigid body motion constraints from two different particles are not imposed at the same velocity nodes. However, in numerical calculations, the overlapping of particles could happen. For solving this problem, Singh, Joseph and coauthors [13] suggested a modified repulsive force model in which the particles are allowed to come arbitrarily close and even to overlap slightly each other. When conflicting rigid body motion constraints from two different particles are applied onto a velocity node, then the constraint from the particle that is closer to that node is used. A repulsive force is only applied when the particles overlap each other. In both models, a short-range repulsion force between particles which are in near contact or slightly overlap is introduced, and the choice of stiffness parameters is very important. In the general case, there is no rigorous theory to determine the ideal values of these parameters. So, they can be referred to as parametric models. For avoiding the difficult work of choosing the parameters in collision models, Diaz-Goano and Minev [19] proposed another collision treatment which leads to a nonparametric model. In this model, first of all there is a check if the separation distance between the particles is larger than a given threshold value calculated as a function of the particle diameters and the mesh resolution. If the distance is less than this value, then the repulsive force is calculated

iteratively so that both particles move along the line that passes through the centres of mass of both particles such that the minimum distance is still maintained.

Following those models proposed by Glowinski, Joseph, Singh and coauthors, we examine another collision model with a new definition of short range repulsive forces which cannot only prevent the particles from getting too close, it can also deal with the case of overlapping to each other when numerical simulations bring the particles very close due to unavoidable numerical truncation errors. For the particle–particle collisions, the repulsive force is determined as

$$\mathbf{F}_{i,j}^{P}=\begin{cases}0 & \text{for } d_{i,j}>R_i+R_j+\rho\\ \dfrac{1}{\varepsilon_P}(\mathbf{X}_i-\mathbf{X}_j)(R_i+R_j+\rho-d_{i,j})^2 & \text{for } R_i+R_j\leqslant d_{i,j}\leqslant R_i+R_j+\rho\\ \dfrac{1}{\varepsilon'_P}(\mathbf{X}_i-\mathbf{X}_j)(R_i+R_j-d_{i,j}) & \text{for } d_{i,j}\leqslant R_i+R_j\end{cases} \tag{7}$$

where R_i and R_j are the radius of the ith and jth particle, $\mathbf{X}_i$ and $\mathbf{X}_j$ are the coordinates of the centres, $d_{i,j}=|\mathbf{X}_i-\mathbf{X}_j|$ is the distance between the mass centres, ρ is the range of the repulsive force (usually $\rho=0.5\sim2.5\Delta h$, Δh is the mesh size), ε_P and ε'_P are small positive stiffness parameters for particle–particle collisions. If the fluid is sufficiently viscous, and $\rho\simeq\Delta h$ as well as ρ_i/ρ_{f} are of order 1 (ρ_i is the density of the ith particle, ρ_{f} is the fluid density), then we can take $\varepsilon_P\simeq(\Delta h)^2$ and $\varepsilon'_P\simeq\Delta h$ in the calculations. For the particle-wall collisions, the corresponding repulsive force reads

$$\mathbf{F}_{i}^{W}=\begin{cases}0 & \text{for } d'_i>2R_i+\rho\\ \dfrac{1}{\varepsilon_W}(\mathbf{X}_i-\mathbf{X}'_i)(2R_i+\rho-d'_i)^2 & \text{for } 2R_i\leqslant d'_i\leqslant 2R_i+\rho\\ \dfrac{1}{\varepsilon'_W}(\mathbf{X}_i-\mathbf{X}'_i)(2R_i-d'_i) & \text{for } d'_i\leqslant 2R_i\end{cases} \tag{8}$$

where $\mathbf{X}'_i$ is the coordinate vector of the centre of the nearest imaginary particle P'_i located on the boundary wall Γ w.r.t. the ith particle, $d'_i=|\mathbf{X}_i-\mathbf{X}'_i|$ is the distance between the mass centres of the ith particle and the centre of the imaginary particle P'_i. ε_W is a small positive stiffness parameter for particle–wall collisions, usually it can be taken as $\varepsilon_W=\varepsilon_P/2$ and $\varepsilon'_W=\varepsilon'_P/2$ in the calculations. Then, the total repulsive forces (i.e. collision forces) exerted on the ith particle by the other particles and the walls can be expressed as follows:

$$\mathbf{F}'_i=\sum_{j=1,j\neq i}^{N}\mathbf{F}_{i,j}^{P}+\mathbf{F}_i^{W} \tag{9}$$

Remark 1

The new expression for the repulsive forces in Equations (7) and (8) has three ranges which correspond to three different cases, i.e. no collision, coming very close, and slightly overlapping, respectively. Since large overlapping is not a correct physical phenomenon and can also cause diverging calculations, only small overlapping is allowed in actual computations.

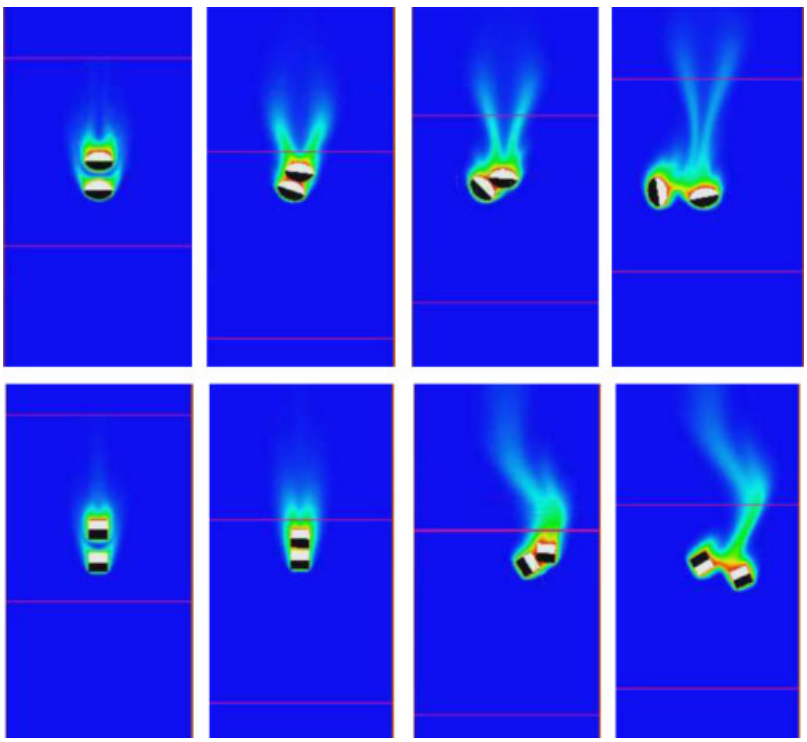

Figure 1. Visualization for collisions between two ellipses (up) and two squares (down), starting from left to right.

Remark 2

Most available collision models are derived only for particles with circular shape rather than for general shaped cases. However, the collision model of Equations (7) and (8) can be extended in principle to the case of particles with more complex shape, provided that a careful calculation for the distance between particles is made. The detailed description of a corresponding collision model for general shapes will be presented in a forthcoming paper. Figure 1 shows preliminary results of two ellipses and two squares handled by the collision model in Equations (7) and (8). It is worthy to be noted that Pan, Glowinski and Joseph have also begun to discuss collisions between noncircular and nonspherical particles in their recent papers [20, 21].

3. NUMERICAL SOLUTION PROCESS

3.1. Multigrid fictitious boundary method

In order to solve the coupled system of fluid and particles, see Equations (1)–(5), we consider an explicit fictitious boundary approach. In that context, the fluid part, the explicit calculation of forces and the movement of the particles are treated in a subsequent manner, which is

obvious in contrast to the (semi-) implicit approaches of Glowinski, Joseph and coauthors [2] as well as of Patankar *et al.* [22]. The strategy for solving the above coupled equations by the explicit fictitious boundary approach can be summarized as follows:

1. Given the positions and velocities of the particles, solve the fluid equations (1) in the corresponding fluid domain involving the position of the particles for the boundary conditions.
2. Calculate the corresponding hydrodynamic forces and the torque acting on the particles by using Equation (4), and compute the collision forces by Equation (9).
3. Solve Equation (3) to get the translational and angular velocities of the particles, and then obtain the new positions and velocities of the particles by Equation (5).
4. Use Equation (6) to set the new fluid domain and boundary conditions, and then advance to solve for the new velocity and pressure of the fluid phase as described in step 1.

If we directly implement this strategy to solve the coupled system of fluid and particles, we will find that its realization is not straightforward because the fluid domain always changes with time and is unknown prior to the solution. Equation (1) governing the fluid motion is active only in the fluid domain Ω_{f}, while the shape and extent of the fluid domain will change in time when the particles move. As mentioned before, the Arbitrary Lagrangian Eulerian (ALE) technique has to remesh the fluid domain which may distort the computational mesh, and complex and very expensive remeshing work may arise. In order to avoid this time-consuming deformation and remeshing step, we adopt a simple and efficient extension of the fictitious boundary method (FBM) for the simulation of particulate flows (see Reference [8]).

This FBM method is based on a multigrid FEM background grid which covers the whole computational domain Ω_T and which is allowed to be stationary or adapted in time (see Section 4.1 concerning aspects of mesh adaptivity for the FBM). It starts with a coarse mesh which may already contain many of the geometrical details of Ω_i $(i=1,\ldots,N)$, and it employs a fictitious boundary indicator (see Reference [8]) which sufficiently describes all fine-scale structures of the particles with regard to the fluid–particle matching conditions of Equation (6). Then, all fine-scale features of the particles are treated as interior objects such that the corresponding components in all matrices and vectors are unknown degrees of freedom which are implicitly incorporated into all iterative solution steps (see Reference [9]). Hence, by making use of Equation (6), we can perform calculations for the fluid in the whole domain Ω_T. The considerable advantage of the multigrid FBM is that the total mixture domain Ω_T does not have to change in time, and can be meshed only once. The domain of definition of the fluid velocity $\mathbf{u}$ is extended according to Equation (6), which can be seen as an additional constraint to the Navier–Stokes equations (1), i.e.

$$\nabla \cdot \mathbf{u} = 0 \quad \text{for } \mathbf{X} \in \Omega_T \tag{10a}$$

$$\rho_{\mathrm{f}} \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \sigma = 0 \quad \text{for } \mathbf{X} \in \Omega_{\mathrm{f}} \tag{10b}$$

$$\mathbf{u}(\mathbf{X}) = \mathbf{U}_i + \omega_i \times (\mathbf{X} - \mathbf{X}_i) \quad \text{for } \mathbf{X} \in \bar{\Omega}_i, \ i = 1, \ldots, N \tag{10c}$$

For the study of interactions between the fluid and the particles, the calculation of the hydrodynamic forces acting on the moving particles is very important. From Equation (4), we can see that the surface integrals on the wall surfaces of the particles should be conducted for the calculation of the forces $\mathbf{F}_i$ and T_i. However, in the presented multigrid FBM method, the shapes of the wall surface of the moving particles are implicitly imposed in the fluid field. If we reconstruct the shapes of the wall surface of the particles, it is not only a time consuming work, but also the accuracy is only of first-order due to a piecewise constant interpolation from our indicator function. For overcoming this problem, we perform the hydrodynamic force calculations using a volume based integral formulation. To replace the surface integral in Equation (4), we introduce a function α_i

$$\alpha_i(\mathbf{X}) = \begin{cases} 1 & \text{for } \mathbf{X} \in \Omega_i \\ 0 & \text{for } \mathbf{X} \in \Omega_T \backslash \Omega_i \end{cases} \tag{11}$$

where $\mathbf{X}$ denotes the coordinates. The importance of such a definition of the parameter can be seen from the fact that the gradient of α_i is zero everywhere except at the wall surface of the ith particle, and equals to the normal vector $\mathbf{n}_i$ defined on the grid, i.e. $\mathbf{n}_i = \nabla\alpha_i$ (see also Reference [10]). Then, the hydrodynamic forces acting on the ith particle can be computed by

$$\mathbf{F}_i = -\int_{\Omega_T} \sigma \cdot \nabla\alpha_i \, \mathrm{d}\Omega, \quad T_i = -\int_{\Omega_T} (\mathbf{X} - \mathbf{X}_i) \times (\sigma \cdot \nabla\alpha_i) \, \mathrm{d}\Omega \tag{12}$$

Through Equation (12), we can calculate $\mathbf{F}_i$ and T_i via the volume integral over the whole domain Ω_T instead of the surface integral over the wall surface of the ith particle in Equation (4). The integral over each element covering the whole domain Ω_T is evaluated with a standard Gaussian quadrature of corresponding high order. Since the gradient $\nabla\alpha_i$ is nonzero only near the wall surface of the ith particle, thus the volume integrals need to be computed only in one layer of mesh cells around the ith particle which leads to a very efficient treatment.

3.2. Discretization in space and time

One of the common solution approaches for Equation (10) is a separate discretization in space and time. We first semi-discretize in time by one of the usual methods known from the treatment of ordinary differential equations, such as the Backward Euler-, Crank–Nicolson- or Fractional-step-θ-scheme (see References [3, 23]). Then, we obtain a sequence of generalized stationary Navier–Stokes equations with prescribed boundary values for every time step. Here, we use the Fractional-step-θ-scheme [3] which is a strongly A-stable time stepping approach and possesses the full smoothing property which is important in the case of rough initial or boundary data. In each time step, we obtain a nonlinear saddle point problem which has to be discretized in space. For the spatial discretization, we choose a finite element approach: A regular quadrilateral mesh T_h for the whole computational domain Ω_T is introduced, where h is used as a parameter characterizing the maximum width of the elements of T_h. To obtain the fine mesh T_h from a coarser mesh T_{2h}, we simply connect opposite midpoints. In the fine grid T_h, the old midpoints of T_{2h} become vertices. We choose the nonconforming $\tilde{Q}_1/Q_0$ element

pair which uses rotated bilinear shape functions for the velocity spanned by $\langle x^2 - y^2, x, y, 1\rangle$ in 2D and piecewise constants for the pressure in the cells. The nodal values are the mean values of the velocity over the element edges and the mean values of the pressure over the elements, rendering this approach nonconforming. This nonconforming $\tilde{Q}_1/Q_0$ element pair has several advantageous features. It satisfies the Babuška–Brezzi condition without any additional stabilization, and the stability constant is independent of the shape and size of the elements. In particular on meshes containing highly stretched and anisotropic cells, the stability and the approximation property are always satisfied. In addition, it admits simple upwind strategies which lead to matrices with certain M-matrix properties. A detailed discussion of these aspects can be found in References [23, 24].

3.3. Discrete projection scheme

For solving the discrete nonlinear problems after time and space discretizations, we have to take the following points into account, i.e. treatment of the nonlinearity, treatment of the incompressibility, and complete outer control like convergence criteria for the overall outer iteration, number of splitting steps, convergence control, embedding into multigrid, etc. In general, there are (at least) two possible approaches for solving the discrete problems [25].

One is the so-called full Galerkin schemes: first, we treat the nonlinearity by an outer nonlinear iteration of fixed point- or quasi-Newton type or by linearization via extrapolation in time, and then we obtain linear subproblems (Oseen equations) which can be solved by a direct coupled or a splitting approach separately for velocity and pressure. Typical schemes are preconditioned GMRES-like or multigrid solvers based on smoothers/preconditioners of type Vanka, SIMPLE or local pressure Schur complement [23]. The disadvantage of these approaches is the high numerical cost for small time steps which are typical for particulate flows. Another possibility are the projection type schemes: first, we split the coupled problem and obtain definite problems in $\mathbf{u}$ (Burgers equations) as well as in p (Pressure-Poisson problems). Then we treat the nonlinear problems in $\mathbf{u}$ by an appropriate nonlinear iteration or linearization technique while optimal multigrid solvers are used for the Poisson-like problems. Classical schemes belonging to this class are the Chorin and van Kan projection schemes and the discrete projection method, all of them are well suited for dynamic configurations which require small time steps (see Reference [26]).

In this paper, based on the latter approach combined with multigrid methods, we adopt the discrete projection method (DPM) as special variant of the more general multigrid pressure Schur complement (MPSC) schemes to solve the discrete nonlinear problems after time and space discretization. A detailed description of DPM and MPSC schemes has been presented in Reference [23]. We first perform as outer iteration a fixed point iteration, applied to the fully nonlinear momentum equations. Then, in the inner loop, we solve the corresponding velocity equations involving linear transport-diffusion problems. Finally, the pressure is updated via a Pressure Poisson-like problem, and the corresponding velocity field is adjusted. Since every time step requires the solution of linearized Burgers equations and Poisson-like problems, an optimized multigrid approach is used. The most important components are matrix–vector multiplication, smoothing operator and grid transfer routines (prolongation and restriction) for the underlying FEM spaces which have been realized in FeatFlow (see Reference [23] for the details).

3.4. Algorithmic details

The whole algorithm of the multigrid FEM and the fictitious boundary method (FBM) for particulate flow can be summarized as follows:

1. Given (initial) particle positions $\mathbf{X}_i$ and velocity $\mathbf{U}_i$ in the overall domain Ω_T. Next, we assume that we have finished the calculations at time t_n.
2. Set the fictitious boundary conditions by using Equation (10c) with the 'old' particle positions $\mathbf{X}_i^n$ and the velocity $\mathbf{U}_i^n$ at time t_n.
3. By using the FBM and implementing the discrete projection scheme, solve the fluid equations of Equations (10a) and (10b) to get the fluid velocity $\mathbf{u}^{n+1}$ and the pressure p^{n+1} at time t_{n+1} on the full computational domain Ω_T.
4. Calculate the hydrodynamic forces $\mathbf{F}_i^{n+1}$ and T_i^{n+1} exerted on every particle $(i=1,\ldots,N)$ by using the volume integration formulation of Equation (12).
5. When two particles come too close, the time step has to be reduced. Then, we adopt several substeps with $\Delta t_k = K/\Lambda$ $(k=1,\ldots,\Lambda, \Lambda$ is the number of substeps calculations, $K = t_{n+1} - t_n)$ for calculating the collisions and updating the particle positions and velocities during the collisions. Set $\mathbf{U}_i^{n,0} := \mathbf{U}_i^n$ and $\mathbf{X}_i^{n,0} := \mathbf{X}_i^n$.
6. Determine the number of substep calculations Λ by

$$\Lambda = \begin{cases} 1 & \text{if } (d_{i,j})_{\min} \geqslant (R_i + R_j)_{\max} \\ \text{MIN}\left\{10, 1 + \text{MAX}\left(\dfrac{|d_{i,j} - R_i - R_j|}{\varrho}\right)\right\} & \text{if } (d_{i,j})_{\min} < (R_i + R_j - \varrho)_{\max} \end{cases} \tag{13}$$

   where ϱ is the maximum penetration distance to be allowed (maximal overlapping range).
7. By using the Newton–Euler equations of Equation (5) to calculate the motion of the solid particles, we obtain the new interim particle position $\mathbf{X}_i^{n+1/2,k}$ and velocity $\mathbf{U}_i^{n+1/2,k}$ as well as the new angular velocity ω_i^{n+1} and angle θ_i^{n+1} by

$$\mathbf{U}_i^{n+1/2,k} = \mathbf{U}_i^{n,k} + \left(\frac{\Delta M_i \mathbf{g}}{M_i} + \frac{\mathbf{F}_i^n + \mathbf{F}_i^{n+1}}{2M_i}\right)\Delta t_k \tag{14}$$

$$\mathbf{X}_i^{n+1/2,k} = \mathbf{X}_i^{n,k} + \left(\frac{\Delta M_i \mathbf{g}}{M_i} + \frac{\mathbf{F}_i^n + \mathbf{F}_i^{n+1}}{2M_i}\right)(\Delta t_k)^2 \tag{15}$$

$$\omega_i^{n+1} = \omega_i^n + \left(\frac{T_i^n + T_i^{n+1}}{2\mathbf{I}_i}\right)K, \quad \theta_i^{n+1} = \theta_i^n + \left(\frac{\omega_i^n + \omega_i^{n+1}}{2}\right)K \tag{16}$$

8. Use the collision model of Equations (7) and (8) to calculate the repulsive forces $(\mathbf{F}_i')^{n+1,k}$ with the interim particle position $\mathbf{X}_i^{n+1/2,k}$.
9. Update the particle positions and the velocity by the repulsive forces to obtain the new particle position $\mathbf{X}_i^{n+1,k}$ and the velocity $\mathbf{U}_i^{n+1,k}$ at time t_{n+1} by

$$\mathbf{U}_i^{n+1,k} = \mathbf{U}_i^{n+1/2,k} + \frac{(\mathbf{F}_i')^{n+1}}{M_i}\Delta t_k, \quad \mathbf{X}_i^{n+1,k} = \mathbf{X}_i^{n+1/2,k} + \frac{(\mathbf{F}_i')^{n+1}}{M_i}(\Delta t_k)^2 \tag{17}$$

10. Set $\mathbf{U}_i^{n,k+1} := \mathbf{U}_i^{n+1,k}$ and $\mathbf{X}_i^{n,k+1} := \mathbf{X}_i^{n+1,k}$ if $k < \Lambda$, and repeat the steps 7–10.
11. Set $\mathbf{U}_i^{n+1} := \mathbf{U}_i^{n+1,\Lambda}$ and $\mathbf{X}_i^{n+1} := \mathbf{X}_i^{n+1,\Lambda}$.
12. Advance to the next new time step, set $t_n := t_{n+1}$ and repeat the steps 2–12.

3.5. Data structures for large number of particles

A challenging situation is the case with large numbers of rigid particles, for instance, the range of 10 000 and more. Indeed, a huge part of the CPU time is required for the force calculations and the fictitious boundary settings with increasing number of particles, while the cost for the Navier–Stokes solver is more or less independent of the number of particles (see Table II). To make it possible that the presented multigrid FBM method is able to simulate the particulate flows with such a large numbers of particles, special techniques are required for the multigrid FBM which decrease the required CPU time. These hierarchical techniques include the following aspects:

1. Find the maximum controlling area of each element; on the coarsest mesh level, check how many particles are inside of the controlling area of each element.
2. On the next finer mesh level, there is no need to search again for every particle, just use the information obtained from the previous coarser level. Since every element of the next finer level would be also within the previous coarse mesh level, search only those particles which are within the previous coarse level.
3. Since all midpoints of the previous coarser level become vertices of the next finer level, use this information for the midpoints of the previous level mesh already obtained and assign them directly to the corresponding vertex point on the next finer mesh level.
4. The vertices or midpoints are possibly occupied by more than one particle (for example, in the case of overlapping), the values for velocities in these points are obtained by the average values of the velocities of those particles who occupy the same points.
5. On the finest mesh, use a new array in the code and assign special values to this array: if a nodal point is not occupied by any particle, its value is set to 0; if a nodal point is occupied by the ith particle, its value is set to i. This array helps to reduce the CPU time for the volume integration of the force calculation.

To evaluate these techniques regarding the CPU time when simulating particulate flows with large numbers of particles, we analyse the cases of 10–100 000 particles falling down in a rectangular cavity filled with an incompressible Newtonian viscous fluid with and without these hierarchical techniques. In Table I, 'NVT' denotes the number of vertices, 'NMT' the number of edges (midpoints), 'NEL' the number of elements and 'NEQ' the total number of unknowns. Table II shows the typical CPU time needed (COMPAQ EV6, 666 MHz) for

Table I. Details of the meshes used in the test calculations.

Level	NVT	NMT	NEL	NEQ
1	1406	27 888	13 824	69 600
2	55 777	111 072	55 296	277 440
3	222 145	443 328	221 184	1 107 840
4	886 657	1 771 392	884 736	4 427 520
5	3 542 785	7 081 728	3 538 944	17 702 400

Table II. Typical CPU time needed in particle flow calculations (one time step) without (top table) and with (bottom table) the time reducing hierarchical techniques.

No. of particles	= 10			= 100			= 1000		
Level	$L=3$	$L=4$	$L=5$	$L=3$	$L=4$	$L=5$	$L=3$	$L=4$	$L=5$
NSE part	24	123	574	20	106	626	22	110	521
Force part	5	20	80	44	176	731	443	1771	7101
Particle part	1	6	26	2	9	43	21	83	332
Total time	30	149	680	66	291	1400	486	1964	7954
Storage (MByte)	4.8	19.5	78.0	4.8	19.5	78.0	4.8	19.5	78.0

No. of particles	= 10			= 1000			= 100 000		
Level	$L=3$	$L=4$	$L=5$	$L=3$	$L=4$	$L=5$	$L=3$	$L=4$	$L=5$
NSE part	16	77	330	16	77	336	14	67	262
Force part	0.1	0.4	1.5	0.1	0.5	1.8	0.2	0.7	2.8
Particle part	1	6	30	2	9	40	622	665	616
Total time	17	84	362	18	87	378	635	733	882
Storage (MByte)	4.5	18.5	74.2	4.6	18.6	74.7	5.7	20.0	75.7

one time step based on the described algorithms with and without the above techniques. The size of computer memory (in MByte) required for each case is also listed in the Table II, in which 'NSE part' means the time for the Navier–Stokes solver, 'Force part' for the calculation of the hydrodynamic forces acting on the particles, 'Particle part' for the fictitious boundary setting and the calculation of the particle–particle and particle–wall collisions. We can see the linear relation between CPU and storage cost w.r.t. the mesh size due to the optimized multigrid components. Moreover, if the hierarchical techniques (or time reducing hierarchical techniques) are not used, the CPU time for the force calculations, the fictitious boundary setting and the calculation of the collisions will significantly grow with increasing the number of particles and mesh refinement. After adopting the hierarchical techniques, the CPU time for the calculation of 100 000 particles is much less than for the calculation of 1000 particles without these techniques. Moreover, the computer memory storage required for both cases is not significantly increased. However, the CPU cost is (still) increasing for many particles and requires further improvements of the algorithmic details. It can be expected that together with more advanced collision models and more efficient data structures, calculations with even 1 000 000 particles and more is possible on modern PC.

4. NUMERICAL EXPERIMENTS

We present test configurations of benchmark character to evaluate and validate the presented methodology. First of all, for the validation of hydrodynamic force calculations based on the volume integration of Equation (12), a careful comparison between the results obtained by the presented method and a standard body-fitted computation is performed for two configurations of two-dimensional flow around a circular body in a channel. The aim is to use the body fitted computation as reference in order to assess the suitability and accuracy of the proposed method. Then, flows with one rotating and moving particle are examined to validate the

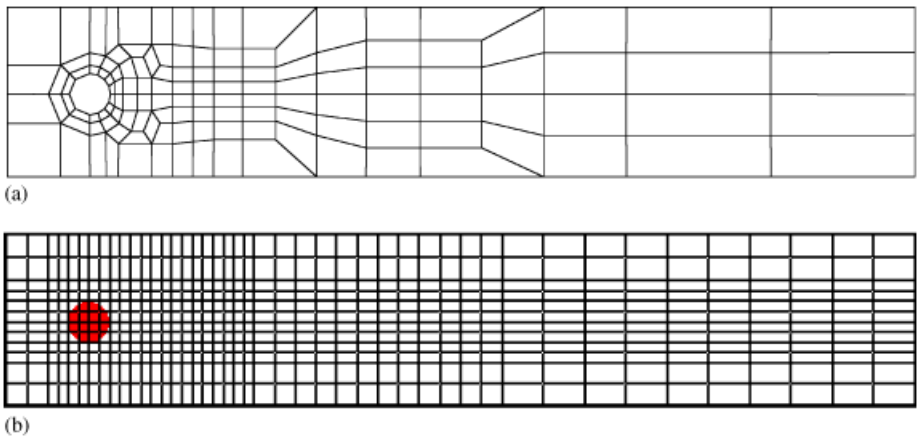

Figure 2. Different coarse meshes adopted for the simulations of flow around a fixed circular cylinder: (a) body-fitted mesh (Level = 1); and (b) Cartesian mesh (Level = 2) for the multigrid FBM.

calculated angular and translational velocities by the presented multigrid FBM. Finally, several cases with two and many particles are analysed to check the quality of the collision model and of the efficiency of the FBM for simulation of particulate flows with large numbers of particles. The time step employed in all calculations is controlled automatically via an implicit estimation of the local truncation error of fluid velocities and pressure (see Reference [23] for the details). In all simulations, we prescribe the geometrical and fluid quantities in dimensionless form.

4.1. Flow around a fixed circular cylinder

In this subsection, we consider the benchmark case of flow around a fixed circular cylinder in a channel as described in Reference [27]. Figure 2 shows a body-fitted mesh around the circular cylinder as well as a Cartesian mesh (rectangular or rectilinear grid) for the FBM, the coloured area shows the region occupied by the cylinder. The shown coarse meshes are successively refined by connecting opposite midpoints. The channel height is $H = 0.41$, the cylinder diameter $D = 0.1$. The centre point of the cylinder is located at $(0.2, 0.2)$. The Reynolds number is defined by $Re = \bar{U}D/\nu$ with the mean velocity $\bar{U} = 2U(0, H/2, t)/3$. The kinematic viscosity of the fluid is given by $\nu = \mu_f/\rho_f = 10^{-3}$ and its density $\rho_f = 1$. The inflow profiles are parabolic $U(0, Y, t) = 6.0\bar{U}Y(H - Y)/H^2$ with different $\bar{U} = 0.2$ and $\bar{U} = 1.0$ such that the resulting Reynolds numbers are $Re = 20$ (steady case) and $Re = 100$ (nonsteady case), respectively. Table III gives the details for these meshes after several global refinements. The meaning of 'Level' is the number of refinements, 'NVT' the number of vertices, 'NMT' the number of edges (midpoints), 'NEL' the number of elements. The total number of unknowns 'NEQ' is $2 \times \text{NMT} + \text{NEL}$ due to the nonconforming FEM data structures. Compared to the body-fitted mesh, in the case of the fixed Cartesian rectilinear mesh, now the cylinder shape is formed by the nodal points which cover the cylinder, instead by a mesh line. 'VEF' means the ratio of the effective cylinder area covered by the fixed mesh with respect to the real cylinder area. We can see that from Level $\geqslant 4$ on, we get an acceptable shape definition.

Table III. Details for sequentially refined meshes used for simulations of flow around a fixed circular cylinder.

	Body-fitted mesh				Cartesian mesh				
Level	NVT	NMT	NEL	NEQ	NVT	NMT	NEL	NEQ	VEF (%)
1	156	286	130	702	161	292	132	716	63.662
2	572	1092	520	2704	585	1112	528	2752	95.493
3	2184	4264	2080	10 608	2225	4336	2112	10 784	95.493
4	8528	16 848	8320	42 016	8673	17 120	8448	42 688	97.482
5	33 696	66 976	33 280	167 232	34 241	68 032	33 792	169 856	99.472
6	133 952	267 072	133 120	667 264	136 065	271 232	135 168	677 632	99.721
7	534 144	1 066 624	532 480	2 665 728	542 465	1 083 136	540 672	2 706 944	99.814

Table IV. Drag and lift coefficients for flow around a fixed circular cylinder with $Re = 20$.

	Body-fitted mesh		Cartesian mesh	
Level	Drag coeff. C_d	Lift coeff. C_l	Drag coeff. C_d	Lift coeff. C_l
3	5.5810	0.004834	5.3303	0.008090
4	5.5608	0.009112	5.4115	0.006530
5	5.5657	0.010163	5.4958	0.010070
6	5.5718	0.010473	5.5405	0.009799
7	5.5755	0.010525	5.5578	0.010147
Reference	$C_d = 5.5795$		$C_l = 0.010618$	

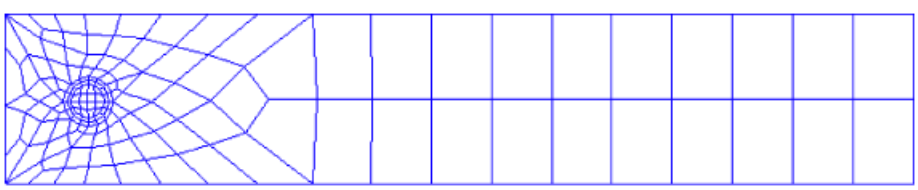

Figure 3. Specially adapted mesh used in the simulation of flow around a fixed circular cylinder (Level = 1).

We first test the steady case of Reynolds number $Re = 20$. Table IV presents drag and lift coefficients by using the two different meshes. From the table, we can see that all results are convergent with respect to mesh refinement, and the case of the fixed rectangular mesh can reach almost the same results as for the body-fitted mesh, especially when the 'area ratio' VEF is greater than 95%. The corresponding reference results for this benchmark problem are also listed in the table for comparison. As can be seen, there is a good agreement between the presented method and the reference computations.

For the simple mesh in Figure 2(b), if we want to get more accurate results for C_d and C_l, the mesh level should be greater than 6 (this means that NEL will be 540 672)! For increasing accuracy and reducing NEL, we test an adapted mesh as shown in Figure 3, which is refined near the wall surface of the cylinder. Table V shows the corresponding results for

Table V. Drag and lift coefficients for flow around a fixed circular cylinder with $Re = 20$ by using the specially adapted mesh.

Level	NEL	C_d	C_l
3	2344	5.569	0.0002
4	9376	5.575	0.0014
5	37 504	5.572	0.0081
6	150 016	5.578	0.0102

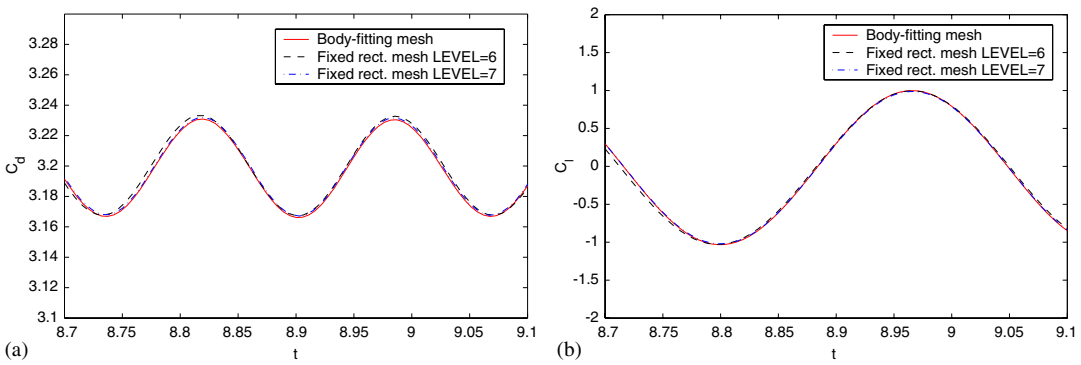

Figure 4. Drag C_d and lift coefficient C_l of one period for flow around a fixed circular cylinder with $Re = 100$. Solid line is for body-fitted mesh, dash line for fixed Cartesian mesh Level = 6, dot-dash line for fixed Cartesian mesh Level = 7.

C_d and C_l. We can see that this specially adapted mesh on level 3 with only NEL = 2344 can already reach accurate results. Therefore, the accuracy can be increased while at the same time the amount of computational work will not be raised if special adaptive mesh techniques are adopted.

Next we perform unsteady flow calculations with the higher Reynolds number $Re = 100$ using the same meshes as in Figure 2. Similarly, the body-fitted mesh is used for the reference calculation while the fixed Cartesian mesh is for the presented FBM. In the reference calculation, Level = 7 of the body-fitted mesh is used, while for the case of the fixed Cartesian mesh, two meshes, Level = 6 and 7, are employed. Figure 4 presents the drag and lift coefficients during one typical period of van Karman vortex shedding when the flow is fully developed. From this figure, we can see that the FBM on the fixed Cartesian mesh can reach almost the same results as in the case of the body-fitted mesh. The drag C_d and lift coefficient C_l, obtained on the coarser fixed Cartesian grid with Level = 6 and on the finer Level = 7, are more or less equal to those obtained from the reference calculation (see Figure 4).

4.2. *A moving circular cylinder in a channel*

The next level of difficulty is introduced when the cylinder is in motion relative to the fixed background mesh. In order to be able to use the reference body-fitted computations for comparison, the calculations will be carried out in a reference frame moving with the cylinder in the case of the body-fitted mesh, whereas a reference frame fixed to the channel

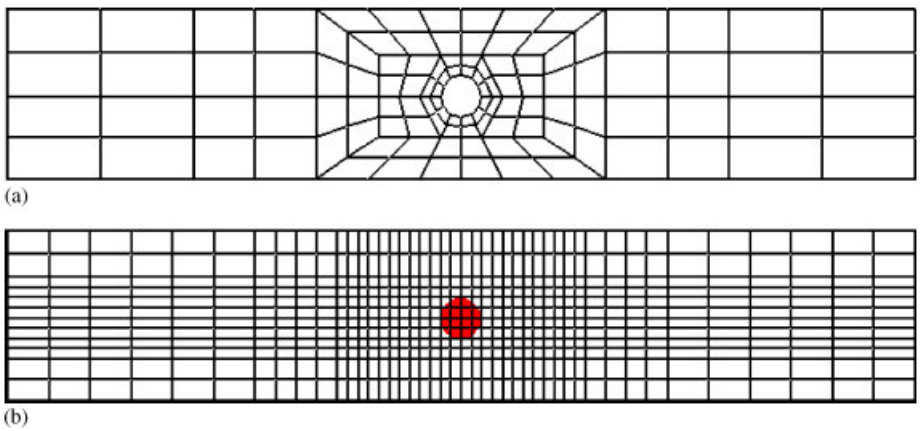

Figure 5. Different coarse meshes adopted for a moving circular cylinder in a channel: (a) body-fitted mesh (Level = 2); and (b) fixed Cartesian mesh (Level = 2).

Table VI. Details of the sequentially refined meshes used for a moving cylinder in a channel.

	Body-fitted mesh				Cartesian mesh			
Level	NVT	NMT	NEL	NEQ	NVT	NMT	NEL	NEQ
1	40	68	28	164	161	292	132	716
2	136	248	112	608	585	1112	528	2752
3	496	944	448	2336	2225	4336	2112	10 784
4	1888	3680	1792	9152	8673	17 120	8448	42 688
5	7360	14 528	7168	36 224	34 241	68 032	33 792	169 856
6	29 056	57 728	28 672	144 128	136 065	271 232	135 168	677 632
7	115 456	230 144	114 688	574 976	542 465	1 083 136	540 672	2 706 944

will be used for the fixed Cartesian mesh. Both cases are equivalent if a velocity $U_m = 2\pi f A \cos(2\pi f t)$, $A = 0.25$, $f = 0.25$ is imposed at the inlet part of the domain, and a slip velocity $U_p = U_m$ is defined on the channel walls in the body-fitted case (see Reference [9] for the details). The cylinder is moved with a prescribed velocity U_m and no-slip velocity conditions are imposed at the two walls, inlet and outlet of the domain in the case of the fixed Cartesian mesh. In Figure 5, the body-fitted mesh is shown for the reference calculation while the fixed Cartesian mesh is taken for the presented FBM. In the reference calculation, the Level = 7 of the body-fitted mesh is used, while for the case of the fixed Cartesian mesh, four meshes on Level = 4 to 7 are employed. Table VI gives the geometrical details of the meshes in Figure 5 with different numbers of refined levels.

Figure 6 illustrates the comparison of the drag coefficient C_d and the lift coefficient C_l between the results of the presented FBM based on the fixed Cartesian mesh and the reference calculation based on the body-fitted mesh. The results calculated from Level = 4 to 7 are all shown together (see Figures 6(a) and (b)). The corresponding coefficients C_d and C_l for one period between $t = 19.79$ and 23.79 are shown in Figures 6(c) and (d), the solid line represents the results of the reference calculation based on the body-fitted mesh at Level = 7,

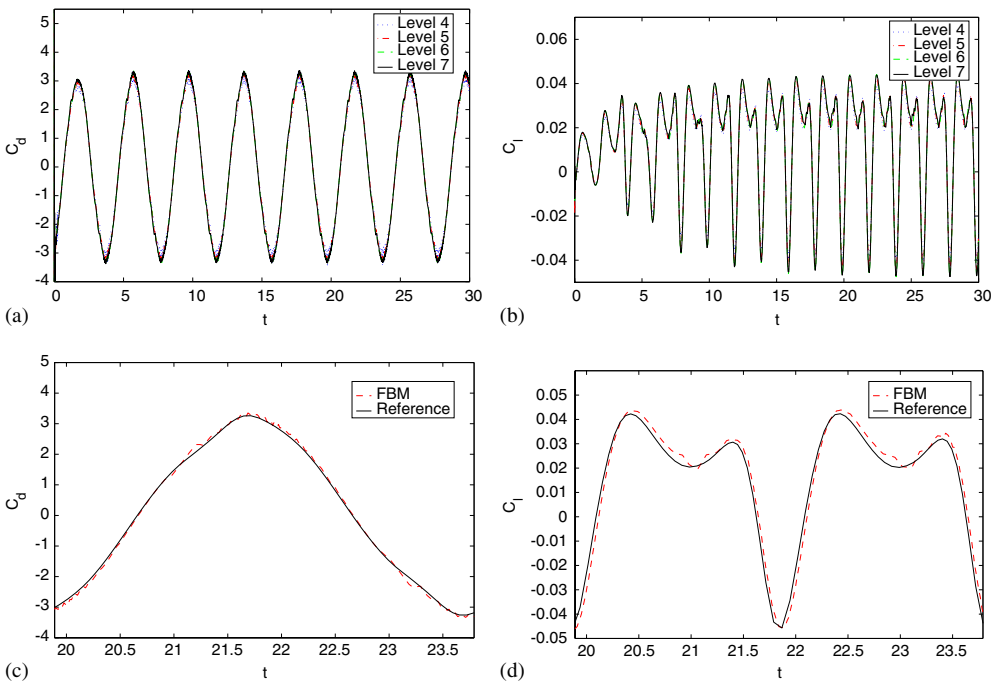

Figure 6. Comparison of C_d and C_l between the FBM and body-fitted meshes for a moving circular cylinder in a channel: (a) C_d of FBM; (b) C_l of FBM; (c) one period of C_d; and (d) one period of C_l.

while the dash line denotes the results obtained by the presented FBM based on the fixed Cartesian mesh at Level = 7. We can see that the FBM results agree very well with the reference results, although the FBM results exhibit small oscillations due to the nonaligned cylinder movement in time over the fixed grid points. Compared to the previous case, the grid refinement has more influence onto the accuracy of the results. This is due to the fact that when the cylinder is moving on the fixed background mesh, depending on the number of nodes currently covered by the cylinder, its effective shape may change. However, the effect of this change of shape on the computed forces is quite small.

4.3. One particle in a rotating circular container

Now we consider a particle with radius 1 centred in a circular container of radius 2 to validate the angular velocity calculation by the presented FBM, starting from rest. The boundary condition at the wall of the container imposes a rotation with an angular velocity $\omega = 0.01$. Then, the particle should start rotating with the same angular speed, i.e. the steady solution is a rigid body rotation inside the container (including the particle) with the same angular velocity $\omega = 0.01$. The fluid density and the particle density are taken as $\rho_f = \rho_p = 1$. The mesh for the present calculation has 9281 nodes and 9216 elements. Table VII gives the calculated terminal angular velocity of the particle at steady state with respect to various values of the

Table VII. The angular velocity at steady state for a particle in a rotating container.

Viscosity v	Terminal angular velocity	Time for reaching the steady state
0.001	0.0099185	7000
0.01	0.0099989	600
0.1	0.0099998	60
1.0	0.0099999	10

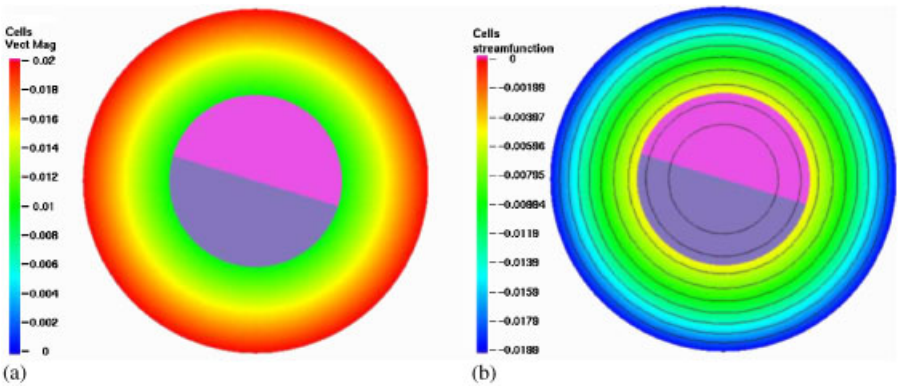

Figure 7. The motion of a particle in a rotating container: (a) velocity; and (b) streamline.

fluid viscosity v. We can see that the terminal angular velocity of the particle approaches the terminal angular speed which matches that of the wall of the container. Figure 7 shows the velocity distribution and the streamline contours at steady state with viscosity $v = 0.01$. Moreover, in Table VII, we also present the time needed for the particle to reach the steady limit with different viscosities. When the viscosity v becomes bigger, the particle can reach the steady state much faster, the terminal angular velocity of the particle is also much closer to the one of the wall of the container.

4.4. *One rotating particle in a shear flow*

For further confirmation of the angular velocity calculation, we consider a single, circular particle placed in the middle between two walls. The radius of the particle is small compared to the distance of the two walls. The channel height is $H = 6$, the width is $W = 4$ and the radius of the circular particle varies from $R \approx 0$ to $R = 1$. The fluid viscosity is set to $v = 0.01$. The fluid density is same as the particle density, i.e. $\rho_f = \rho_p = 1$. The centre point of the particle is located at $(2.0, 3.0)$. The left and right walls are supposed to move vertically with velocities $U = 0.02$ and -0.02 so that, without any particle, the stationary flow would be a linear shear flow between the walls, with uniform vorticity $\Omega = 0.01$. The centre of the particle is fixed, but it can rotate around its centre. The corresponding stationary flow is such that the angular velocity of the particle is $\omega = -\Omega/2 = -0.005$ when the radius of the circular particle is small enough. An equidistant mesh with 222 145 nodes and 221 184 elements is used. Table VIII presents the terminal angular velocity of the particle and the time needed to reach the steady state with different radii of the particles. Figure 8 shows the velocity distribution

Table VIII. Terminal angular velocity at steady state for a rotating particle in a shear flow.

Radius of particle R	Terminal angular velocity	Time reaching the steady state
~ 0	-0.0049993	230
1/20	-0.0049909	240
1/10	-0.0049837	250
1/5	-0.0049584	270
2/5	-0.0048697	310
1	-0.0043148	600

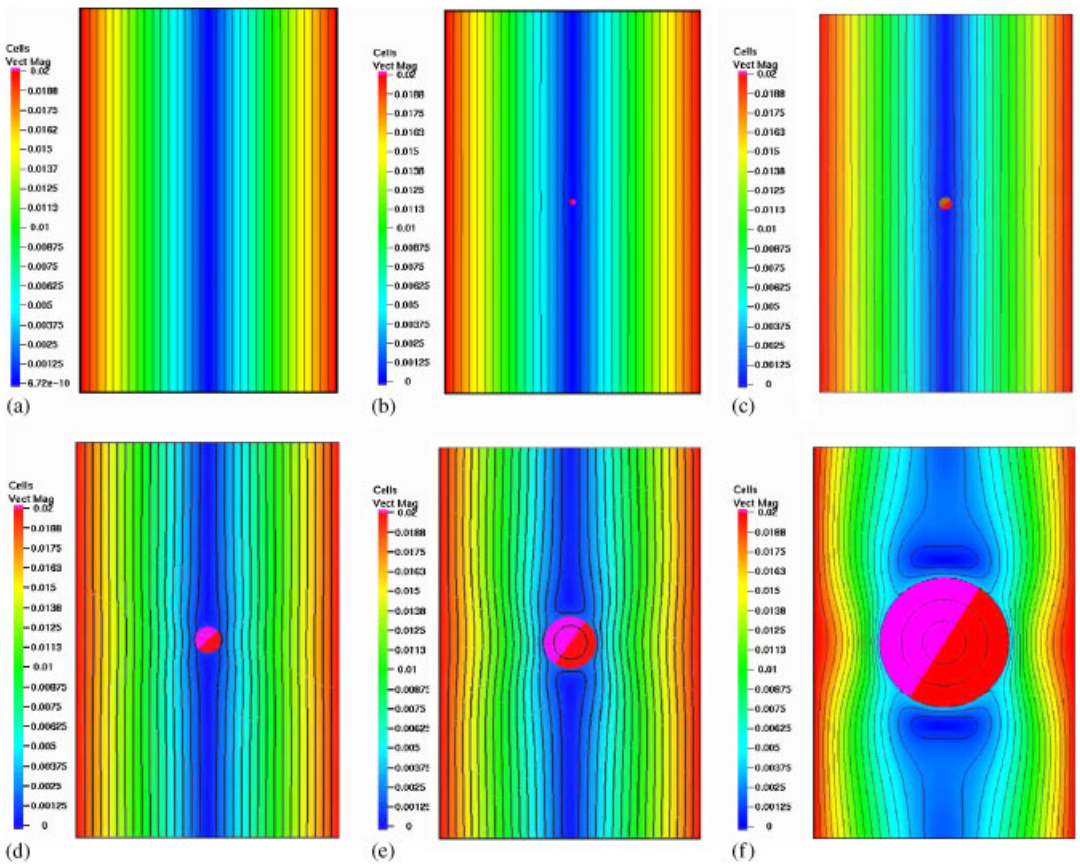

Figure 8. Norm of velocity distribution of a rotating particle in a shear flow: (a) $R \approx 0$; (b) $R = 1/20$; (c) $R = 1/10$; (d) $R = 1/5$; (e) $R = 2/5$; and (f) $R = 1$.

at steady state, and Figure 9 provides the evolution of the particle angular velocity in time for different radii R. It can be seen that the terminal angular velocity of the particle increases until it approximates very well half of the vorticity of the fluid field. When the radius of the circular particle becomes smaller, the terminal angular velocity approaches closer to this

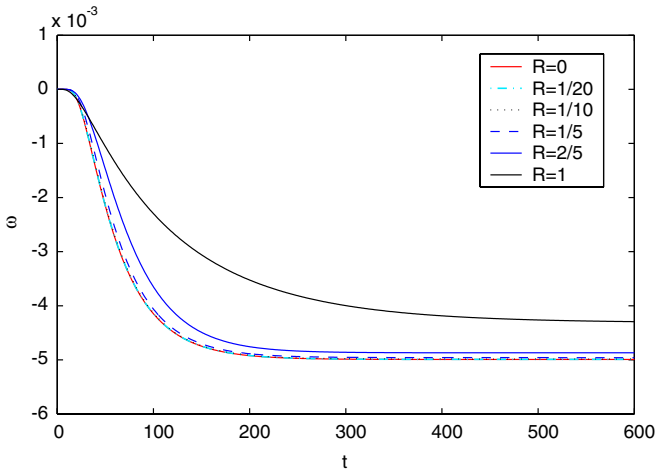

Figure 9. Angular velocity vs time for a rotating particle in a shear flow.

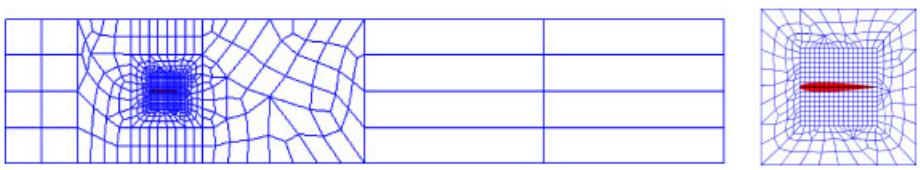

Figure 10. Coarse mesh for flow around a rotating NACA0012 airfoil (left is the global mesh, right is a local zoom covering the airfoil at Level = 1).

value due to less disturbance of the fluid field, and the particle can also reach the steady state much faster.

4.5. Flow around a rotating airfoil

The rigid particles considered so far have been circular disks. The main goal of the following test problem is to show that the presented FBM can be easily applied to rigid bodies of more complicated shape, still providing accurate results.

We consider incompressible viscous flow around a NACA0012 airfoil that has a fixed centre of mass and is free to rotate due to hydrodynamical forces (see the description in References [2, 3]). Figure 10 shows the fixed Cartesian coarse mesh which is locally adapted, the coloured area is the region occupied by the airfoil at rest; this coarse mesh is successively refined by connecting opposite midpoints. Table IX gives the geometrical details for the refined meshes. In the table, 'Level' is the number of global refinements, 'NVT' the number of vertices, 'NMT' the number of edges (midpoints), 'NEL' the number of elements, 'NEQ' the total number of unknowns. In this fixed Cartesian mesh, the airfoil shape is formed by the nodal points which are inside the airfoil, instead of a mesh line. 'VEF' means the ratio of the

Table IX. Details for sequentially refined meshes used for flow around a rotating NACA0012 airfoil.

Level	NVT	NMT	NEL	NEQ	VEF (%)
1	625	1220	596	3036	95.018
2	2441	4824	2384	12 032	97.393
3	9649	19 184	9536	47 904	99.769
4	38 369	76 512	38 144	191 168	100.14
5	153 025	305 600	152 576	763 776	99.973

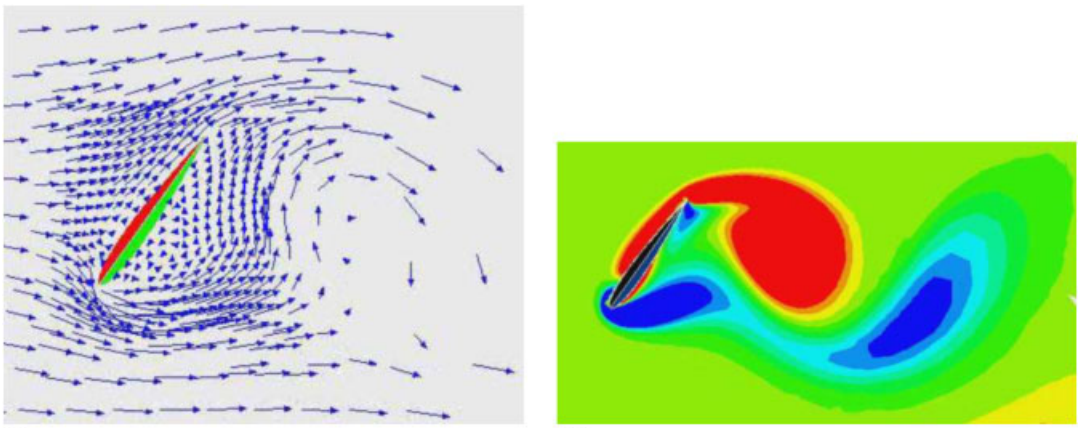

Figure 11. Snapshot for flow around a rotating NACA0012 airfoil with $Re = 101$: (a) vector field; and (b) vorticity.

effective airfoil area covered by the fixed Cartesian mesh compared with the real airfoil area. We can see that from Level $\geqslant 3$ on, we get an acceptable shape approximation.

The surrounding region is the rectangle $(0, 20) \times (0, 4)$. The characteristic length, namely the airfoil length, is 1.0089304 and the fixed centre of mass of the airfoil is at $(0.420516, 2)$. The shape of the NACA0012 is described as follows (for $0 \leqslant X \leqslant 1.0089304$):

$$Y = 0.6 \cdot \{0.2969 \cdot \sqrt{X} + X \cdot [-0.126 + X \cdot [-0.3516 + X \cdot (0.2843 - 0.1015 \cdot X)]]\} \qquad (18)$$

Initial angular velocity and angle of incidence are zero. The density of the fluid is $\rho_f = 1$ and the density of the airfoil is $\rho_p = 1.1$. The viscosity of the fluid is $v_f = 10^{-2}$. The initial condition for the fluid velocity is $\mathbf{u}(0) = 0$ and the boundary conditions are given as $\mathbf{u} = 0$ when $y = 0$ or 4 and $\mathbf{u} = 1$ when $x = 0$ or 20 for $t \geqslant 0$. Here, the Reynolds number is approximately $Re \approx 101$ with respect to the length of the airfoil and the maximum inflow speed [3]. In these simulations, we start the calculation with Level $= 3$, until it reaches a stable periodical motion (here at $t = 42.05$), then we change to Level $= 4$ and continue the calculation based on the initial values obtained from interpolation using the results of Level $= 3$. Similarly, after $t = 77.05$, we change to Level $= 5$ based on the results of Level $= 4$. The flow field and the vorticity are shown in Figure 11. Figure 12 presents the history of the angular velocity ω and

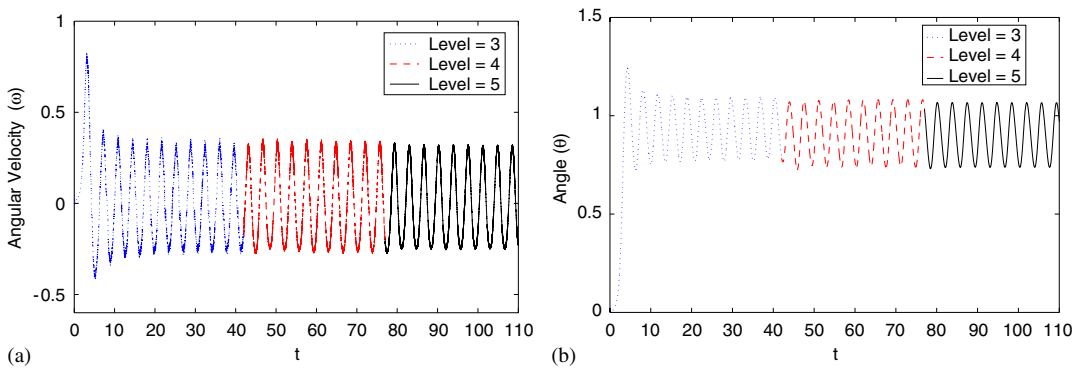

Figure 12. Computed data for flow around a NACA0012 airfoil with $Re = 101$. Dotted line is for Level = 3, dashed line for Level = 4, solid line for Level = 5: (a) history of the angular velocity ω; and (b) history of the angle θ.

the angle θ of the airfoil corresponding to the three different meshes. We observe that the results are in excellent agreement with those obtained by Glowinski, Joseph and coauthors [1, 3] and converge to a mesh independent solution after the flow reaches a periodic motion.

4.6. *Trajectory of a moving spinning ball*

For verifying our multigrid FBM to simulate real particle motion with both free angular velocities and translational velocities, we test a moving spinning ball. We begin the experiment by pushing and spinning the ball impulsively, i.e. for time $t<0$, everything is at rest; for $t=0^+$, the constant velocity $\mathbf{v}(0^+)=\mathbf{v}_0$ and rotational speed $\omega(0^+)=\omega_0$ are specified. Huang [28] has given the analytical result for the trajectory of such a moving spinning ball for $t>0$. For comparing with these results, we set the radius of the ball to $R=1$, fluid viscosity $\nu=1$, fluid density $\rho_f=1$, density of the ball $\rho_p=10$, initial velocity of the ball $U_0=100$, $V_0=0$, initial angular velocity of the ball $\omega_0=50$. The computational domain is a sufficiently large rectangle $(-10,30)\times(-10,20)$. For this test problem, we have chosen a uniform mesh with different sizes, namely $\Delta h=1/8$ for Level = 3 with 77 361 nodes and 76 800 elements, as well as $\Delta h=1/16$ for Level = 4 with 308 321 nodes and 307 200 elements.

Figure 15 shows the computed data for the case $\rho_p=10$ by using the two different meshes with Level = 3 and 4 which include (a) trajectory of the ball until time $t=0.3$, (b) history of the moving ball angle, (c) history of the u-component of the velocity of the moving ball, (d) history of the v-component of the velocity of the moving ball, (e) history of the angular velocity of the moving ball, (f) history of the ratio (%) of the actual area of the moving ball covered by the mesh with respect to the real area of the moving ball, (g) history of the translational kinetic energy of the moving ball, $E_T=0.5\,M\,(u^2+v^2)$, M is the mass of the ball, u and v are the u-component and v-component of the velocity at the centre point of the ball, (h) history of rotational kinetic energy of the moving ball, $E_R=0.5\,\mathbf{I}\omega^2$, $\mathbf{I}$ is the moment of inertia of the ball, ω is the angular velocity of the ball. From these pictures, we can see that the results of Level = 3 are essentially the same as those for Level = 4. The computed

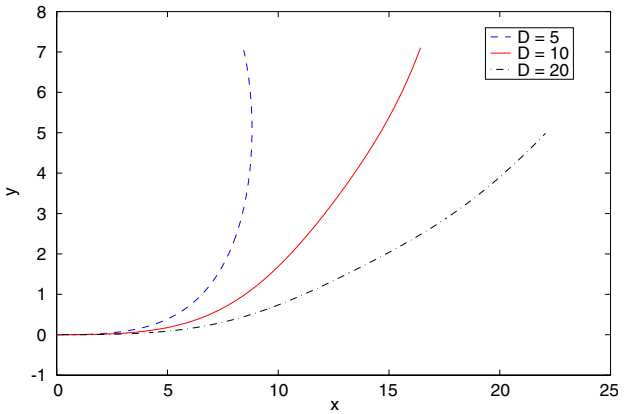

Figure 13. Trajectory for a moving spinning ball with different densities at $t = 0.3$. Dashed line is for $\rho_p = 5$, solid line for $\rho_p = 10$, dot-dashed line for $\rho_p = 20$ by using the Level = 4 mesh.

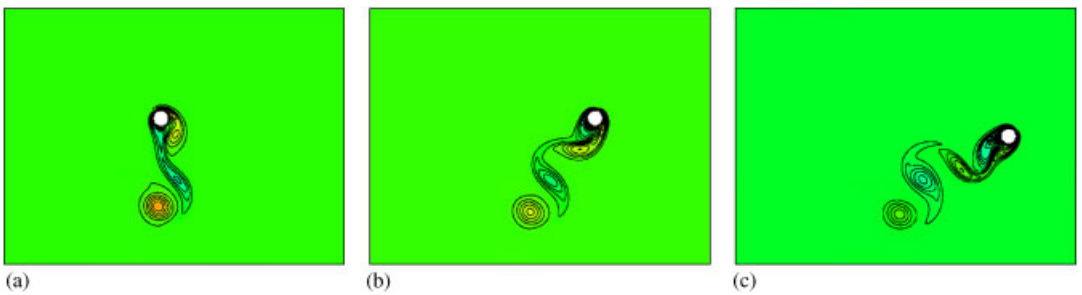

Figure 14. Contour plots of vorticity at $t = 0.3$ for a moving spinning ball: (a) $\rho_p = 5$; (b) $\rho_p = 10$; and (c) $\rho_p = 20$.

results converge with respect to the mesh size and agree very well with the analytical results of Reference [28].

By Newton's 2nd law, we know that if we want to increase the change of direction of the ball, we have to decrease the density of the ball ρ_p. For checking this, we perform two further calculations with $\rho_p = 5$ and 20. Figure 13 provides the trajectories of the ball until $t = 0.3$ for the three cases of $\rho_p = 5$, 10 and 20 by using Level = 4. We can see that when ρ_p decreases, the trajectory of the ball becomes more steep and has a bigger curvature. Figures 14 and 15 finally shows the contour plot of vorticity at $t = 0.3$ for the three different ball densities $\rho_p = 5$, 10 and 20.

4.7. Single disk falling in an incompressible fluid

Next, we consider the numerical simulation of the motion of a circular disk falling in an incompressible Newtonian viscous fluid to further validate our multigrid FBM. The test problem we perform is the following: The computational domain is a channel of width 2 and

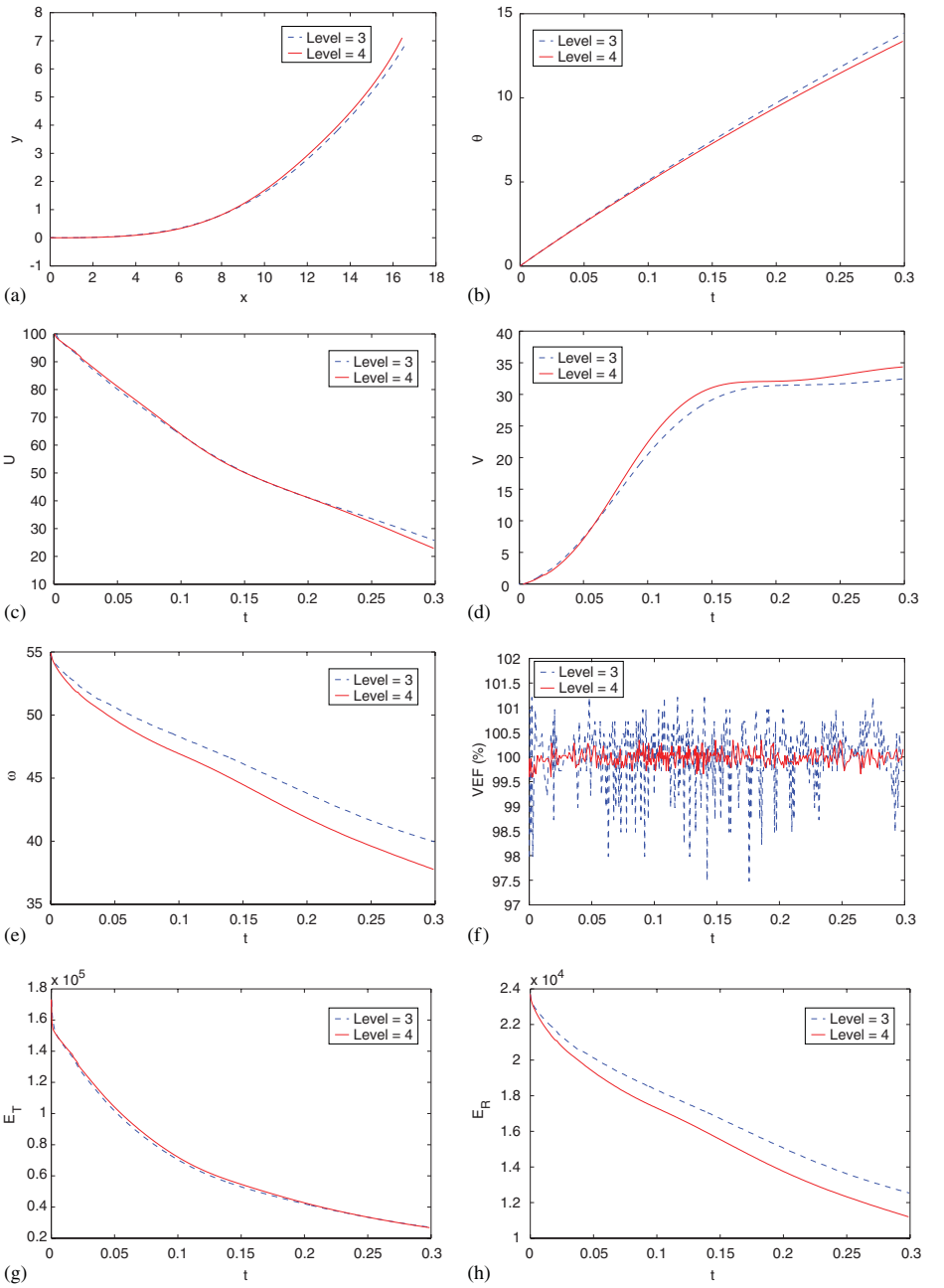

Figure 15. Calculated results for a moving spinning ball with $\rho_p = 10$: (a) trajectory at $t = 0.3$; (b) angle θ with time; (c) u-component with time; (d) v-component with time; (e) angular velocity ω with time; (f) effect volume ratio with time; (g) translational kinetic energy with time; and (h) rotational kinetic energy with time.

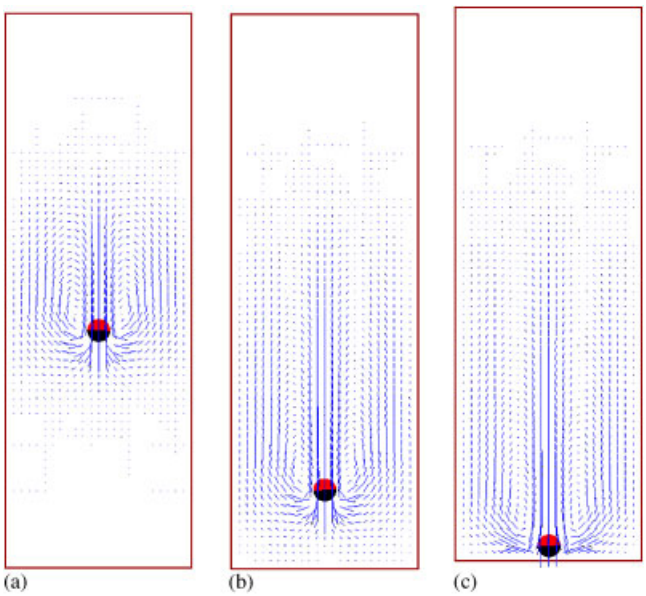

Figure 16. One circular disk falling down in an incompressible fluid: (a) $t = 0.34$; (b) $t = 0.66$; and (c) $t = 0.79$, from left to right.

height 6. A rigid circular disk with diameter $d = 0.25$ and density $\rho_p = 1.25$ is located at $(1, 4)$ at time $t = 0$, and it is falling down under gravity in an incompressible fluid with density $\rho_f = 1$ and viscosity $v = 0.1$. We suppose that the disk and the fluid are initially at rest. The simulation is carried out on two different mesh sizes, i.e. $\Delta h = 1/48$ on Level = 3 with 28 033 nodes and 27 648 elements, as well as $\Delta h = 1/96$ on Level = 4 with 111 361 nodes and 110 592 elements.

Figure 16 shows the velocity field computed with $\Delta h = 1/96$ on Level = 4. We can see that the circular disk quickly reaches a uniform falling velocity until it hits the bottom of the channel. In Figure 17, some of the quantities are plotted in time. These are the y-coordinate of the disk centre, v-component of the translational disk velocity, Reynolds number of the disk defined by $Re = \sqrt{u^2 + v^2} \cdot d \cdot \rho_p / v$, and translational kinetic energy ($E_T = 0.5\, M\, (u^2 + v^2)$, M is the mass of the disk, u and v are the u-component and v-component of the velocity at the centre point of the disk), as well as the rotational kinetic energy ($E_R = 0.5\, \mathbf{I}\omega^2$, $\mathbf{I}$ is the moment of the inertia of the disk, ω is the angular velocity of the disk), and the ratio (%) of the effective area of the disk covered by the underlying fixed mesh compared with the real area of the disk, for the two different mesh sizes on Level = 3 and on Level = 4. It can be seen, as expected, that the results on Level = 4 show less numerical oscillations than on Level = 3. The maximum computed Reynolds numbers are 17.42 for Level = 3 and 17.15 for Level = 4 compared to the result of 17.31 provided by Glowinski [3]. We can see that the results computed on the two different mesh sizes are essentially the same except for the case of the rotational kinetic energy due to its very small values ($< 10^{-3}$) which is easily disturbed by numerical errors.

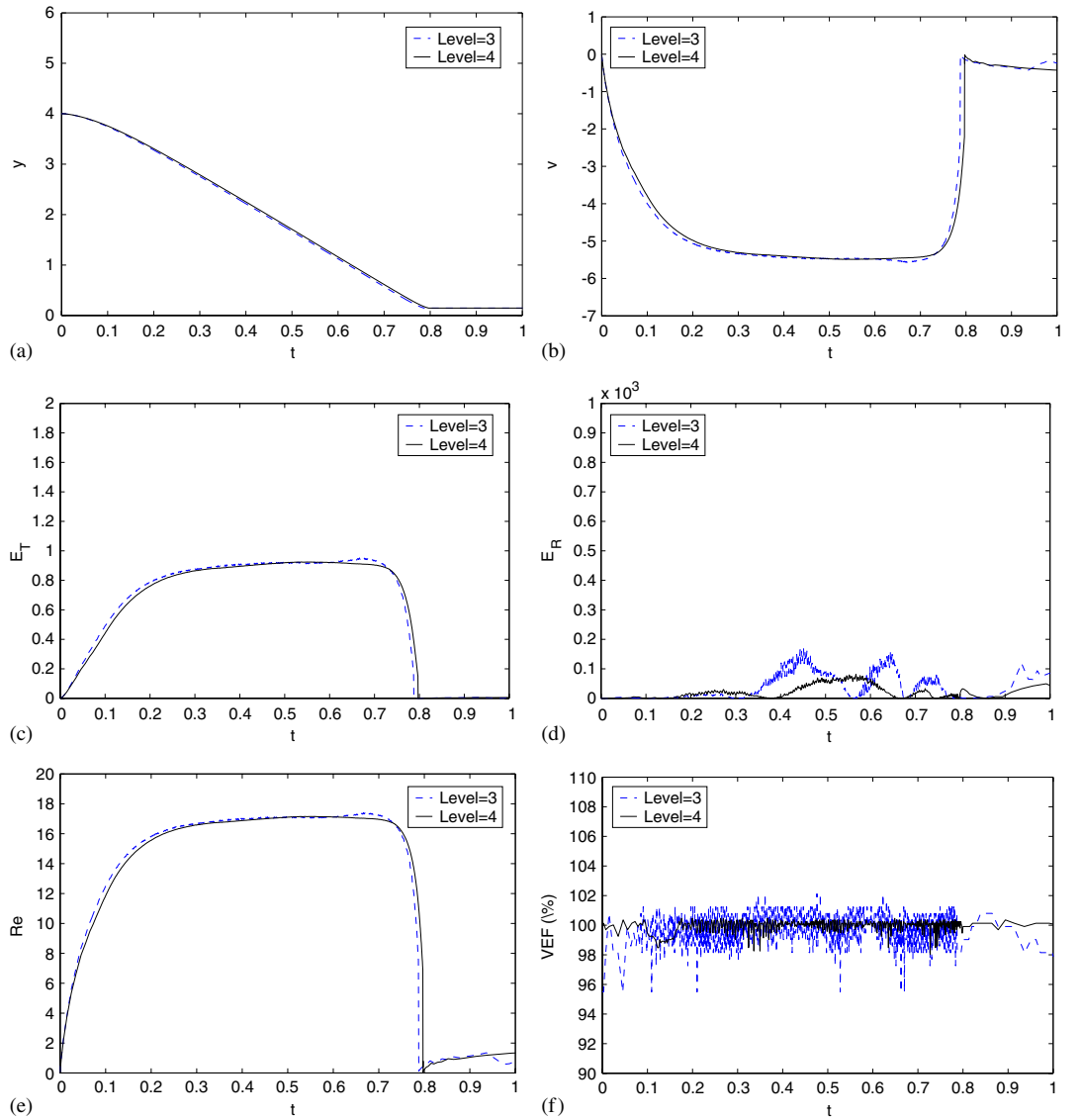

Figure 17. One circular disk falling down in an incompressible fluid ($\rho_p = 1.25$): time histories of: (a) the y-coordinate of the particle centre; (b) v-component of the translational particle velocity; (c) translational kinetic energy; (d) rotational kinetic energy; (e) Reynolds number for the disk; and (f) the effective area ratio of the disk covered by the used equidistant mesh with respect to the real area of the disk, dashed line for $\Delta h = 1/48$ (Level = 3), solid line for $\Delta h = 1/96$ (Level = 4).

4.8. Sedimentation of two circular particles

To examine the complete model, including the prevention of collisions, we study the sedimentation of two circular particles in a two-dimensional channel, comparing the results with

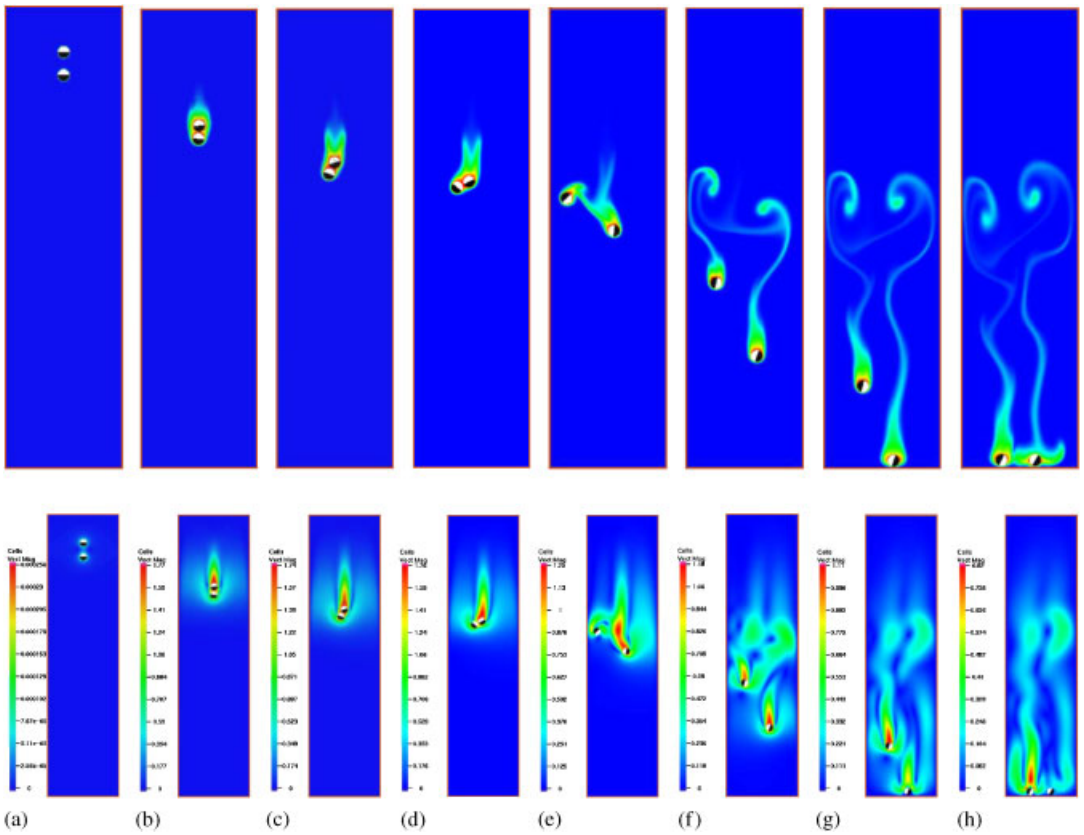

Figure 18. Sedimentation of two circular particles: (a) $t = 0.00$; (b) $t = 1.13$; (c) $t = 1.53$; (d) $t = 1.73$; (e) $t = 2.44$; (f) $t = 4.49$; (g) $t = 6.23$; and (h) $t = 7.57$.

respect to two different mesh sizes and regarding the results in Reference [3]. The computational domain is a channel of width 2 and height 8; the diameter of the particles is $d = 0.2$. The fluid and particle densities are $\rho_f = 1.0$ and $\rho_p = 1.01$, and the fluid viscosity is $v = 0.01$. The simulation is started at $t = 0$ by dropping two particles at the centre of the channel at height 7.2 (No. 1 particle) and 6.8 (No. 2 particle). The simulation is executed for two different mesh sizes, i.e. $\Delta h = 1/48$ on Level $= 3$ with 37 345 nodes and 36 864 elements, as well as $\Delta h = 1/96$ on Level $= 4$ with 148 417 nodes and 147 456 elements, to check the quality of the computed solutions with respect to the mesh size.

It is well known that when two particles are dropped close to each other, they interact by undergoing 'drafting, kissing and tumbling' [29]. Figure 18 shows the visualization of the characteristic behaviour (top figures) of two falling particles, computed with $\Delta h = 1/96$ on Level $= 4$, providing also the corresponding velocity distribution (bottom figures). From these figures, we can see that the particle in the wake (No. 1 particle) falls more rapidly than the particle No. 2 in front since the fluid forces acting on it are smaller. The gap between them decreases, and they almost touch ('kiss') each other at time $t = 1.13$. After touching, the two particles fall together until they tumble ($t = 1.53$) and subsequently they separate from each

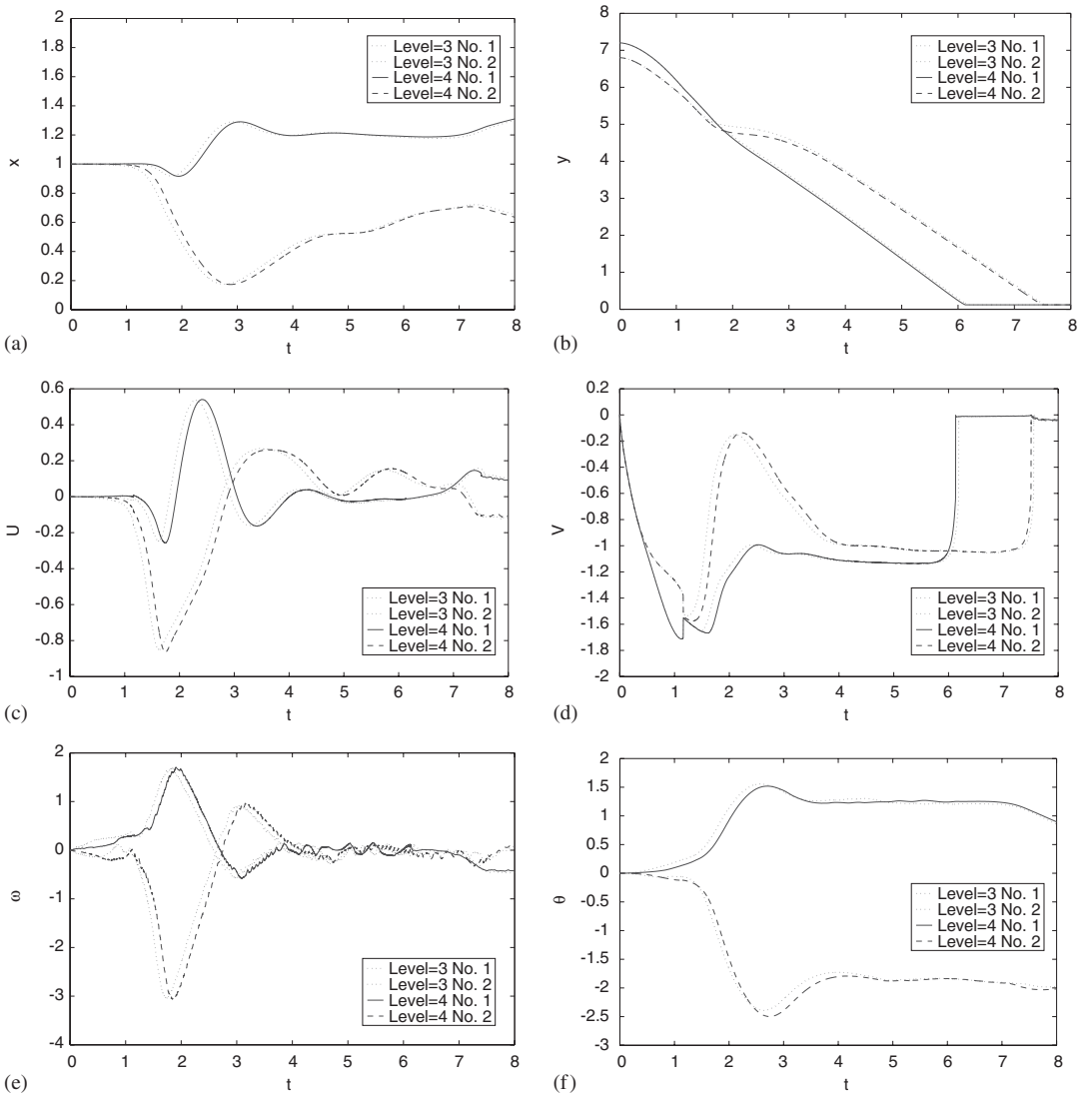

Figure 19. Sedimentation of two circular particles: time histories of: (a) x-coordinate; (b) y-coordinate of the particle centre; (c) u-component; (d) v-component of the translational particle velocity; and of (e) the particle angular velocity; and (f) the particle angle of the particle, obtained for mesh size $\Delta h = 1/48$ (dot-dashed line for No. 1 particle, dotted line for No. 2 particle), and $\Delta h = 1/96$ (solid line for No. 1 particle, dashed line for No. 2 particle).

other ($t = 1.73$). The tumbling of the particles takes place because the configuration, when both are parallel to the flow direction, is unstable. The No. 1 particle is touching first the bottom wall at $t = 6.23$, while the No. 2 particle reaches the bottom wall at $t = 7.57$.

In Figures 19 and 20, several quantities are plotted. These are the time histories of the x-coordinate and y-coordinate of the two particle centres, u-component and v-component of

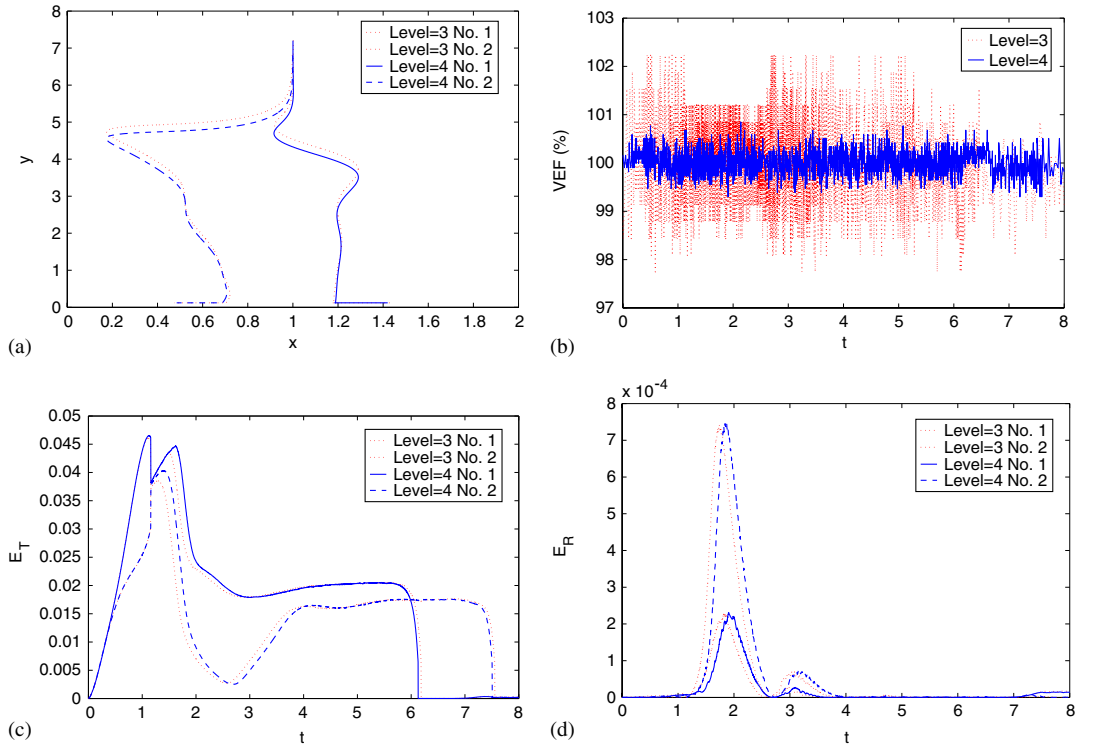

Figure 20. Sedimentation of two circular particles: (a) trajectory at $t = 8$; (b) time history of the ratio (%) of the effective area of the two particles compared with the real area of the two particles; and of (c) the translational as well as of; and (d) the rotational kinetic energy.

the particle translational velocities, the particle angular velocities and the particle angles, the trajectories and the time histories of the ratio (%) of the effective area of the two particles covered by the underlying fixed mesh compared with the real area of the two particles, as well as the translational and rotational kinetic energy of the particles, obtained for the mesh sizes $\Delta h = 1/48$ (Level $= 3$) and $\Delta h = 1/96$ (Level $= 4$). We can see that the results computed on the two different mesh sizes are essentially indistinguishable. Again, we can see that the two particles are kissing each other at $t = 1.13$. After kissing, they tumble, i.e. a manifestation of the instability of a falling long body aligned with the stream. The No. 1 particle has more translational kinetic energy, but less rotational kinetic energy than No. 2 particle. Both particles reach their maximum energy values during the kissing and tumbling. The No. 2 particle rotates much more than the No. 1 particle. It can be seen, as expected, that the accuracy on Level $= 4$ is better than on Level $= 3$. Moreover, all results compare qualitatively well with those presented in References [3, 13, 19, 22]. However, there is still need for further rigorous benchmarks for multiple particles.

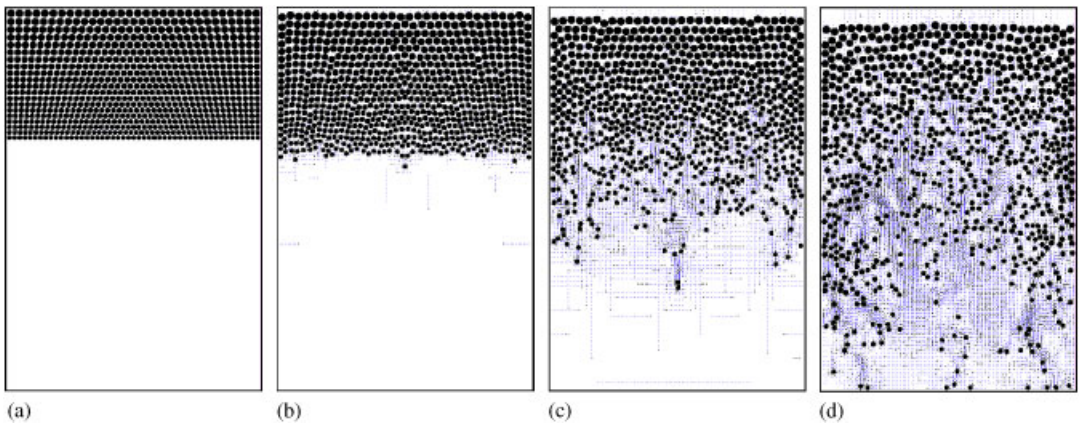

Figure 21. Velocity field for 790 particles with various diameters falling down in a cavity: (a) $t = 0.0$; (b) $t = 0.52$; (c) $t = 1.06$; and (d) $t = 1.60$.

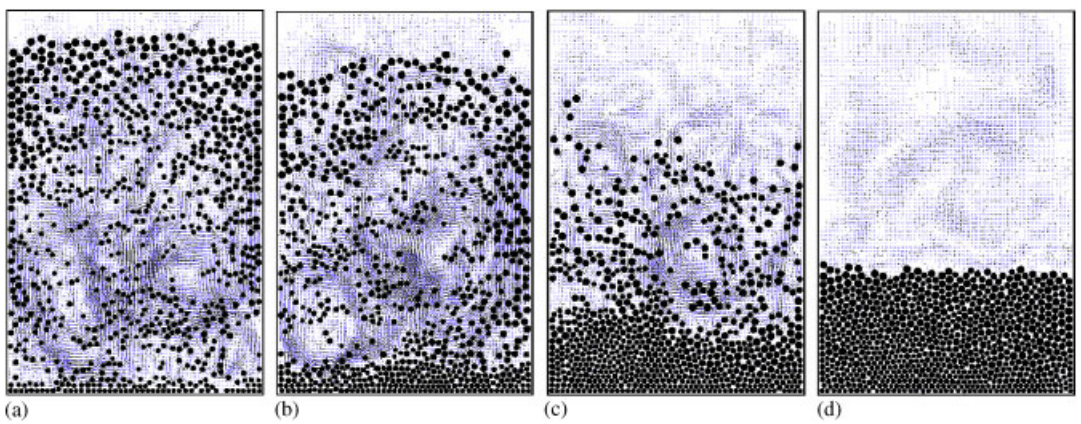

Figure 22. Velocity field for 790 particles with various diameters falling down in a cavity: (a) $t = 2.14$; (b) $t = 2.68$; (c) $t = 4.02$; and (d) $t = 6.19$.

4.9. Sedimentation of 790 circular particles of various size

The following test problems differ significantly from the ones considered so far since a much larger number of rigid particles is used. The aim of the subsequent simulations is to show that the proposed methodology can handle much more complex configurations, too. We have not found corresponding reference values (unless the calculations in Reference [3]), from simulations or experiments so that we can only examine the qualitative behaviour.

The specific problem in this subsection is the sedimentation of 790 circular particles with different diameters falling down in a closed rectangular cavity. The width and height of the cavity are 8 and 12. The 790 particles are placed at the top of the cavity with 20 rows. In each row, the diameter of the particles is the same. The number of particles at the top row

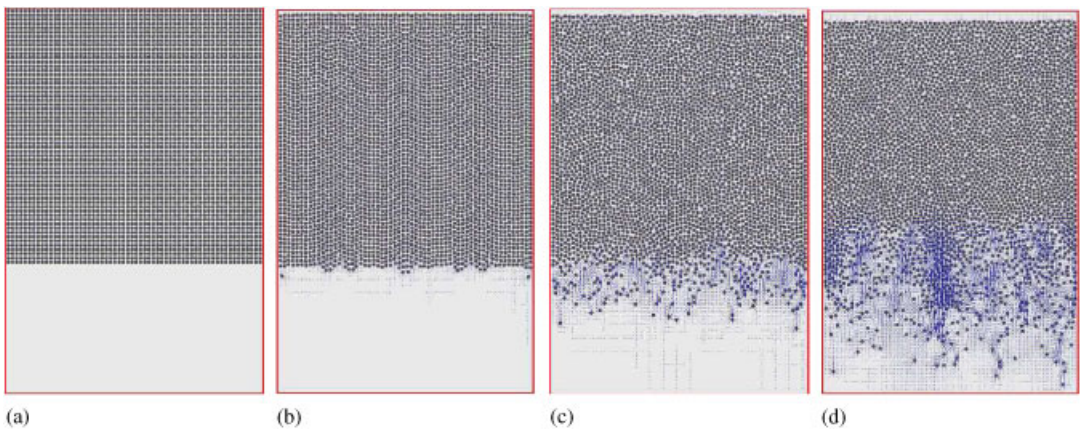

Figure 23. Velocity fields for 3600 particles falling down in a cavity: (a) $t = 0.0$; (b) $t = 0.52$; (c) $t = 1.06$; and (d) $t = 1.60$.

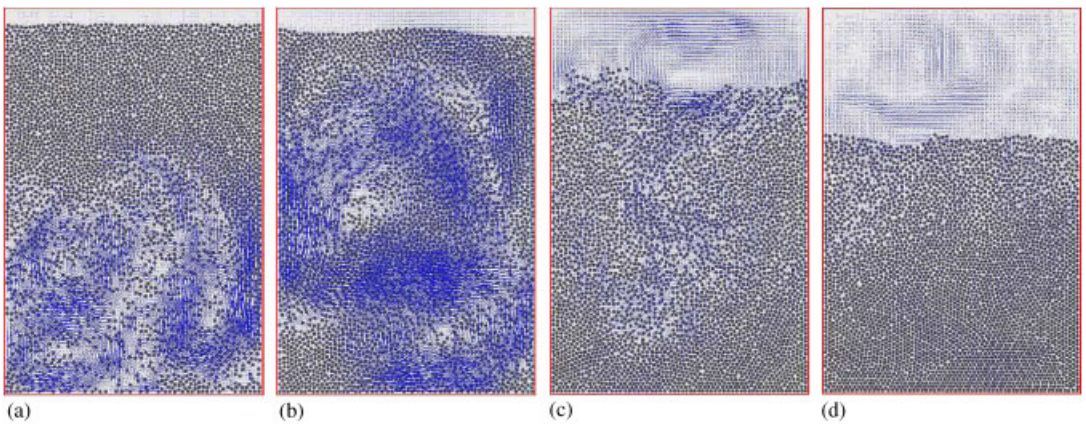

Figure 24. Velocity fields for 3600 particles falling down in a cavity: (a) $t = 2.68$; (b) $t = 4.03$; (c) $t = 7.75$; and (d) $t = 11.18$.

is 30, and then always add 1 particle to the next row. The maximum diameter of the particles is 0.2396, and the minimum diameter is 0.1366. The range of the repulsive force is chosen as $\rho = 0.02604$. The position of the particles at time $t = 0$ is shown in Figure 21(a). The particles and the fluid are at rest at $t = 0$. The density of the fluid is $\rho_f = 1$ and the density of the particles is $\rho_i = 1.1$ $(i = 1, \ldots, 790)$. The viscosity of the fluid is $v = 10^{-2}$. The details of the mesh have been listed in Table I. The parameter ε_P in the collision model has been taken equal to 5×10^{-7}, and $\varepsilon_W = \varepsilon_P/2$, $\varepsilon'_P = \varepsilon_P$, $\varepsilon'_W = \varepsilon_W$. The evolution of the 790 particles is shown in Figures 21 and 22. A Rayleigh–Taylor instability develops in the advancing front between $t = 0.52$ and 1.60. At $t = 1.06$, many complex vortices have been formed which pull

the particles downward and mix each other with particles of different size. Between $t = 2.14$ and 2.68, some stronger eddies are formed which push the particles almost to the top of the cavity. At the end, all particles have settled down to the bottom of the cavity, and the fluid returns to rest.

4.10. Sedimentation of 3600 circular particles

Finally, we consider the sedimentation of 3600 circular particles with identical size falling down in a closed rectangular cavity. The 3600 particles are placed at the top of the cavity with 60 rows, while in each row the number of particles is 60. The diameter of the particles is 0.108. The range of the repulsive force is chosen as $\rho = 0.02502$. The position of the particles at time $t = 0$ is shown in Figure 23(a). The particles and the fluid are at rest at $t = 0$. The density of the fluid is $\rho_f = 1$ and the density of the particles is $\rho_i = 1.1$ $(i = 1, \ldots, 3600)$. The viscosity of the fluid is $\nu = 10^{-2}$ (all quantities in nondimensional form). The mesh and computational domain are the same as in the previous case of 790 particles. The parameter ε_P in the collision model has been taken equal to 10^{-6}, and $\varepsilon_W = \varepsilon_P/2$, $\varepsilon'_P = \varepsilon_P$, $\varepsilon'_W = \varepsilon_W$. The evolution of the 3600 circular particles is shown in Figures 23 and 24. These figures clearly show the development of the Rayleigh–Taylor instability. This instability develops into a fingering and text-book phenomenon, and many symmetry breaking and other bifurcation phenomena including drafting, kissing and tumbling take place at various scales in space and time. Many vortices of different size develop and the phenomenon is clearly 'chaotic'. Finally, the particles settle at the bottom of the cavity and the fluid returns to rest.

5. CONCLUSIONS

We have presented the multigrid FBM for the direct simulation of particulate flows in 2D. Since the method is based on 'simple' extensions of standard Navier–Stokes solvers, the 3D case is quite straightforward and will be part of a forthcoming paper, as soon as we have realized the proposed and analysed methodology in the 3D solvers of our CFD package Feat-Flow [24]. The presented paper has mainly focused on the basic components of the FBM and the numerical examples have benchmarking character to validate the proposed methodology in the context of FEM discretization techniques and multigrid solvers.

The main advantage of the described FBM is that the solid particles, which are allowed to have different shape and size, can move freely through the computational mesh for the fluid part which has not to change in time; however, the mesh can be aligned as we will demonstrate at the end of this article. The proposed volume-based integration for the calculation of the hydrodynamic forces acting on the moving particles is one of the key ingredients of the FBM, and its accuracy has been proven by numerous comparisons between the presented results and corresponding reference results from own computations or from the literature.

As a conclusion, this new approach can be easily incorporated into (almost) all CFD codes without the need for additional (background) meshes for the particles or special interpolation procedures since it only requires changes in the treatment of Dirichlet boundary conditions. On the other hand, the applied (explicit) splitting approach renders the overall scheme to be of first-order only; however, since complex configurations with numerous particles typically require small time steps by physical reasons, the comparison with more implicit scheme, for

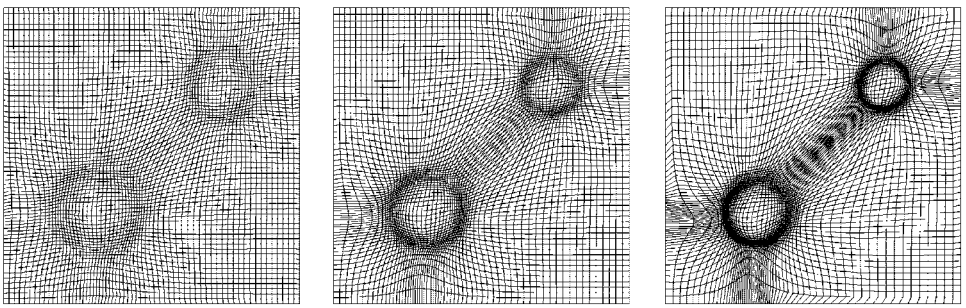

Figure 25. Deformed meshes with different 'grid deformation' parameters.

instance References [3, 22], is not yet clear, particularly with respect to the total efficiency. Hence, it is absolutely necessary to design approximate benchmarks for realistic particulate flows! Another advantage of this new approach is that very different shapes and sizes of particles can be easily included; even discrete coalescence and breakup mechanisms are possible and will be topic of a forthcoming paper.

The modified collision model based on papers by Glowinski, Joseph, Singh and coauthors with a new definition of a short range repulsive force cannot only prevent the particles from getting too close to each other, it can also deal with those cases when the numerical simulation brings the particles very close or even leads to overlapping. Special data structures and time reducing techniques for handling the calculation of large numbers of particles are described to enable the FBM to efficiently solve for many particles. Consequently, one of the next aims is to simulate in 2D up to 10^6 particles on a single PC/workstation while the corresponding 3D module in FeatFlow will be based on a parallel implementation for such a high number of particles. Furthermore, nonNewtonian and viscoelastic fluids ('chaining' instead of 'kissing, drafting, tumbling', see Reference [3]) have to be tackled in the future and will be part of a forthcoming paper.

We finish this paper with a short remark concerning mesh adaptivity for a better approximation of the particles. As we have shown, the use of the multigrid FBM does *not* require to change the mesh during the simulations, although the particles vary their positions. The advantage is that no expensive remeshing has to be performed while a fixed mesh can be used such that in combination with appropriate data structures and fast CFD solvers very high efficiency rates can be reached. However, the accuracy for capturing the particles is only of first-order which might lead to accuracy problems for the explicit calculation of the correct fluid forces acting on the particles. One remedy could be to preserve the 'mesh topology', for instance as generalized tensorproduct or blockstructured meshes, while a local alignment with the particle surfaces is reached via special 'grid deformation' techniques. Preliminary examples for the case of 2 particles can be found in Figure 25 which shows a deformed grid, created from an equidistant Cartesian mesh. While the topology is preserved, only the grid spacing is changed such that the grid points are concentrated near the particle surfaces. This methodology is based on papers by Liao *et al.* [30, 31] and our paper [32] and requires only the solution of additional linear Poisson problems in every time step which means that the additional work is significantly less than the total CFD part (see Table II). Therefore, we plan to present the details of this special adaptive approach in a forthcoming paper.

REFERENCES

1. Glowinski R, Pan TW, Hesla TI, Joseph DD. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow* 1999; **25**:755–794.
2. Glowinski R, Pan TW, 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. *Journal of Computational Physics* 2001; **169**:363–426.
3. Glowinski R. Finite element methods for incompressible viscous flow. In *Handbook of Numerical Analysis*, Ciarlet PG, Lions JL (eds), vol. IX. North-Holland: Amsterdam, 2003; 701–769.
4. Hu HH, Joseph DD, Crochet MJ. Direct simulation of fluid particle motions. *Theoretical and Computational Fluid Dynamics* 1992; **3**:285–306.
5. Hu HH. Direct simulation of flows of solid–liquid mixtures. *International Journal of Multiphase Flow* 1996; **22**:335–352.
6. Maury B. Characteristics ALE method for the unsteady 3D Navier–Stokes equations with a free surface. *International Journal of Computational Fluid Dynamics* 1996; **6**:175–188.
7. Maury B. Direct simulations of 2D fluid–particle flows in biperiodic domains. *Journal of Computational Physics* 1999; **156**:325–351.
8. Turek S, Wan DC, Rivkind LS. The fictitious boundary method for the implicit treatment of Dirichlet boundary conditions with applications to incompressible flow simulations. *Challenges in Scientific Computing*, Lecture Notes in Computational Science and Engineering, vol. 35. Springer: Berlin, 2003; 37–68.
9. Wan DC, Turek S, Rivkind LS. An efficient multigrid FEM solution technique for incompressible flow with moving rigid bodies. *Numerical Mathematics and Advanced Applications, ENUMATH 2003*. Springer: Berlin, 2004; 844–853.
10. Duchanoy C, Jongen TRG. Efficient simulation of liquid–solid flows with high solids fraction in complex geometries. *Computers and Fluids* 2003; **32**:1453–1471.
11. John V. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2002; **40**:775–798.
12. Sigurgeirsson H, Stuart AM, Wan J. Collision detection for particles in a flow. *Journal of Computational Physics* 2001; **172**:766–807.
13. Singh P, Hesla TI, Joseph DD. Distributed Lagrange multiplier method for particulate flows with collisions. *International Journal of Multiphase Flow* 2003; **29**:495–509.
14. Johnson AA, Tezduyar TE. Simulation of multiple spheres falling in a liquid-filled tube. *Computer Methods in Applied Mechanics and Engineering* 1996; **134**:351–373.
15. Maury B. A many-body lubrication model. *Comptes Rendus de l Academie des Sciences Paris. T.*, *Série 1* 1997; **325**:1053–1058.
16. Sommerfeld M. Validation of a stochastic Lagrangian modelling approach for inter-particle collisions in homogeneous isotropic turbulence. *International Journal of Multiphase Flow* 2001; **27**:1829–1858.
17. Asakura K, Asari T, Nakajima I. Simulation of solid–liquid flows in a vertical pipe by a collision model. *Powder Technology* 1997; **94**:201–206.
18. Leal LG. Laminar flow and convective transport processes. *Scaling Principles and Asymptotic Analysis*. Butterworth-Heinemann, Newton: Stoneham, MA.
19. Diaz-Goano C, Minev P, Nandakumar K. In *A Lagrange Multiplier/Fictitious Domain Approach to Particulate Flows*. Margenov W, Yalamov (eds), Lecture Notes in Computer Science, vol. 2179. Springer: Berlin, 2001; 409–422.
20. Pan T-W, Glowinski R, Joseph DD. Simulating the dynamics of fluid-cylinder interactions. *Journal of Zhejiang University SCIENCE* 2005; **6A**:97–109.
21. Pan T-W, Glowinski R, Joseph DD. Simulating the dynamics of fluid-ellipsoid interactions. *Computers and Structures* 2005; **83**:463–478.
22. Patankar NA, Singh P, Joseph DD, Glowinski R, Pan TW. A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow* 2000; **26**:1509–1524.
23. Turek S. *Efficient Solvers for Incompressible Flow Problems*. Springer: Berlin, Heidelberg, New York, 1999.
24. Turek S. **FEATFLOW**—Finite element software for the incompressible Navier–Stokes equations. *User Manual, Release 1.2*, University of Dortmund, 1999.
25. Turek S. A comparative study of time stepping techniques for the incompressible Navier–Stokes equations: from fully implicit nonlinear schemes to semi-implicit projection methods. *International Journal for Numerical Methods in Fluids* 1996; **22**:987–1011.
26. Turek S. On discrete projection methods for the incompressible Navier–Stokes equations: an algorithmical approach. *Computer Methods in Applied Mechanics and Engineering* 1997; **143**:271–288.
27. Schäfer M, Turek S. Benchmark computations of laminar flow around cylinder. In *Flow Simulation with High-Performance Computers II*. Notes on Numerical Fluid Mechanics, Vol. 52, Hirschel EH (ed.). Vieweg: Braunschweig, 1996; 547–566.

28. Huang JY. Trajectory of a moving curveball in viscid flow. *Proceedings of the International Conference on Dynamical Systems and Differential Equations*, Atlanta, U.S.A., 18–21 May 2000; 1–7.
29. Fortes A, Joseph DD, Lundgren T. Nonlinear mechanics of fluidization of beds of spherical particles. *Journal of Fluid Mechanics* 1987; **177**:483–497.
30. Bochev P, Liao G, dela Pena G. Analysis and computation of adaptive moving grids by deformation. *Numerical Methods for Partial Differential Equations* 1996; **12**:489–506.
31. Liao G, Fleitas D, Jiang B. Adaptive grid generation based on least-squares finite element method. *Technical Report*, Department of Mathematics, University of Texas, 2004.
32. Grajewski M, Köster M, Kilian S, Turek S. Numerical analysis and practical aspects of a robust and efficient grid deformation method in the finite element context. 2005, submitted to SISC.

