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

Abstract

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.

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 field and the domain in which it is defined are unknown: it can require a 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 approach is Eulerian 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 [6, 7, 8]. The second type is Lagrangian which is based on a moving mesh following the motion of 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 [9, 10] and Maury [16, 18] have applied the ALE to fluid-particle flows. One big advantage of the Eulerian 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 [27]) 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 [28]. 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.

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 surface. Therefore, the integration of forces only over 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 [4, 12] has been proposed. 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 for 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, a collision model is needed to prevent particles from interpenetrating each other. 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 required element size in Lagrangian approaches required for this decreases, 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: the mesh is to be refined in the gap zone which is computationally expensive. For solving this problem, Glowinski, Joseph, Pan and coauthors [6, 7, 8] have described repulsive force models to prevent the particles from overlapping each other. Singh, Hesla and Joseph [21], following an idea of Joseph, implemented a scheme in which a repulsive force is initiated only when the particles touch or overlap. Their idea was to allow hydrodynamic lubrication forces to act up to the full tolerance of the mesh and allow the particles to pack naturally in equilibrium. There are also many other collision models which are based on physical properties of the particles, for example, lubrication [17] and stochastic collision models [22]. Following such models as proposed by Glowinski, Singh, Joseph 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 models by Glowinski, Joseph and Singh is presented. The fictitious boundary method (FBM) is described in section 3. The proposed volume based integration for hydrodynamic force calculations and its validation are shown together in this section. Section 4 describes the numerical discretization, solution procedure, the algorithmic details and the time reducing techniques for dealing with large numbers of particles. In section 5, several numerical test problems are presented. Flow with one rotating and moving particle is first examined to validate the prediction of angular and translational velocities by the fictitious boundary method. Then, cases of multiple particles are examined to evaluate the collision model and the efficiency of the presented fictitious boundary method for the simulation of particulate flows with large numbers of particles. The paper ends with a conclusion in section 6.

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 ν. 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 $\Omega_f(t)$,

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

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

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

Here $\mathbf{I}$ is the identity tensor, $\mu_f = \rho_f \cdot \nu$, 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

$$M_i \frac{d\,\mathbf{U}_i}{d\,t} = (\Delta M_i)\,\mathbf{g} + \mathbf{F}_i + \mathbf{F}_i', \qquad \mathbf{I}_i \frac{d\,\omega_i}{d\,t} + \omega_i \times (\mathbf{I}_i\,\omega_i) = T_i\,, \tag{3}$$

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 center of mass acting on the ith particle which are calculated by

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

where σ is the total stress tensor in the fluid phase defined by Eq. (2), $\mathbf{X}_i$ is the position of the mass center 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 to 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

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

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

For handling more than one particle, a collision model is needed to prevent the particles from interpenetrating each other. Theoretically, it may be impossible that smooth particle-particle or boundary wall-particle collisions 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 [7, 8, 14]. However, collisions take place 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. In such simulations, numerical problems are likely to occur when particles get very close to each other: the mesh is to be refined in the gap zone, which is computationally expensive.

To handle this problem numerically, several collision models have been proposed in the literature. 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 [7, 8] 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, Joseph et al. [21] 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. In these models, 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, we can refer to the approaches by Glowinski, Joseph and Singh as parametric models. For avoiding the difficult work of choosing the parameters in these collision models, Minev et al. [3] proposed another collision treatment which leads to a nonparametric model. In Minev's 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 centers of mass of both particles such that the minimum distance is still maintained.

Following such models, 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,

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

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 centers, $d_{i,j} = |\mathbf{X}_i - \mathbf{X}_j|$ is the distance between the mass centers, ρ is the range of the repulsive force (usually $\rho = 0.5 \sim 2.5\Delta h$, 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 $\epsilon_P \simeq (\Delta h)^2$ and $\epsilon_P' \simeq \Delta h$ in the calculations. For the particle-wall collisions, the corresponding repulsive force reads,

$$
\mathbf{F}_i^W = \begin{cases} 0, & \text{for } d_i' > 2R_i + \rho, \\ \dfrac{1}{\epsilon_W}(\mathbf{X}_i - \mathbf{X}_i')(2R_i + \rho - d_i')^2, & \text{for } 2R_i \le d_i' \le 2R_i + \rho, \\ \dfrac{1}{\epsilon_W'}(\mathbf{X}_i - \mathbf{X}_i')(2R_i - d_i'), & \text{for } d_i' \le 2R_i, \end{cases} \tag{8}
$$

where $\mathbf{X}_i'$ is the coordinate vector of the center 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 centers of the ith particle and the center of the imaginary particle P_i'. ϵ_W is a small positive stiffness parameter for particle-wall collisions, usually it can be taken as $\epsilon_W = \epsilon_P/2$ and $\epsilon_W' = \epsilon_P'/2$ 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,

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

Remark 1: The new expression for the repulsive forces in Eqs. (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.

Remark 2: Most available collision models are derived only for particles with circular shape, but not for general shaped cases. However, the collision model of Eqs. (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. Fig. 1 shows preliminary results of two ellipses and two squares handled by the collision model in Eqs. (7) and (8).
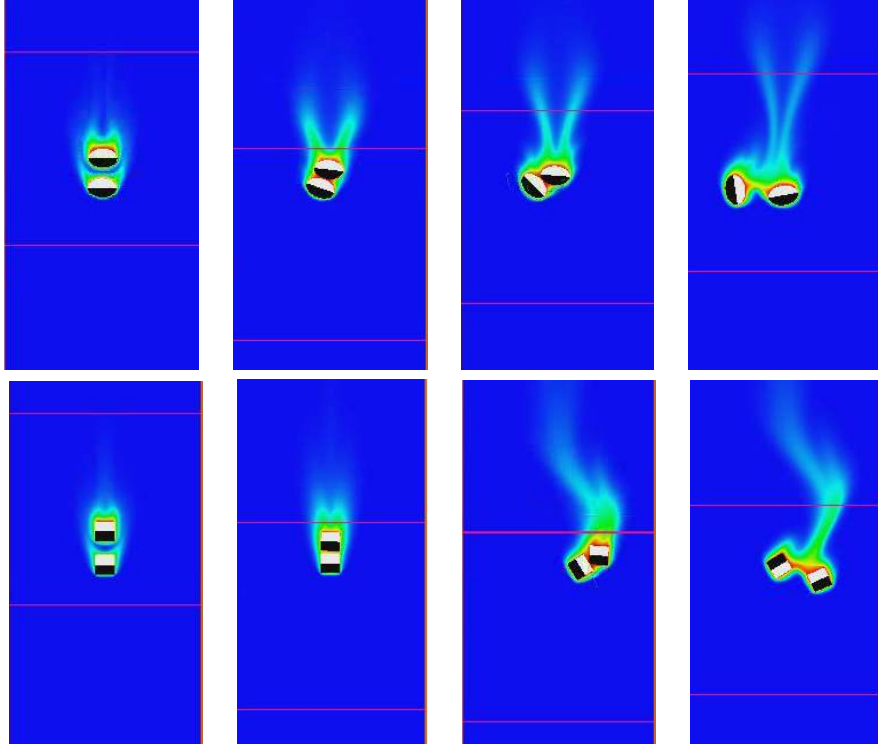

Figure 1: Visualization for two ellipses and two squares, starting from left to right

3   Multigrid Fictitious Boundary Method

In contrast to the (semi-) implicit approaches of Glowinski, Joseph and coauthors [7] as well as of Patankar, Singh, Joseph, Glowinski and Pan [19], we describe an "explicit" approach which treats the fluid part, the (explicit) calculation of forces and the movement of the particles in a subsequent manner. Then, the strategy for solving above equations governing the coupled system of fluid and particles 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 Eq. (4), and compute the collision forces by Eq. (9).

3) Solve Eq. (3) to get the translational and angular velocities of the particles, and then obtain the new positions and velocities of the particles by Eq. (5).

4) Use Eq. (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. Eq. (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 [27]).

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 the next sections 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 [27]) which sufficiently describes all fine-scale structures of the particles with regard to the fluid-particle matching conditions of Eq. (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 [28]). Hence, by making use of Eq. (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 Eq. (6), which can be seen as an additional constraint to the Navier-Stokes equations (1), i.e.,

$$
\begin{cases}
\nabla \cdot \mathbf{u} = 0 & (a) & \text{for} \quad \mathbf{X} \in \Omega_T, \\[2ex]
\rho_f \left(\dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \sigma = 0 & (b) & \text{for} \quad \mathbf{X} \in \Omega_f, \\[2ex]
\mathbf{u}(\mathbf{X}) = \mathbf{U}_i + \omega_i \times (\mathbf{X} - \mathbf{X}_i) & (c) & \text{for} \quad \mathbf{X} \in \bar{\Omega}_i, \ i = 1, \ldots, N.
\end{cases}
\tag{10}
$$

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 Eq. (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 Eq. (4) we introduce a function α_i,

$$
\alpha_i(\mathbf{X}) = \begin{cases}
1 & \text{for} \quad \mathbf{X} \in \Omega_i, \\[2ex]
0 & \text{for} \quad \mathbf{X} \in \Omega_T \setminus \Omega_i,
\end{cases}
\tag{11}
$$

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 [4]). Then, the hydrodynamic forces acting on the ith particle can be computed by

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

Through Eq. (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 Eq. (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 non-zero 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 (see the numerical section). For the validation of these force calculations in the multigrid FBM based on the volume integration of Eq. (12), a careful comparison between the results obtained by the presented method and a standard body-fitted computation has been performed for two configurations for 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. For more details, see [27] and [28].

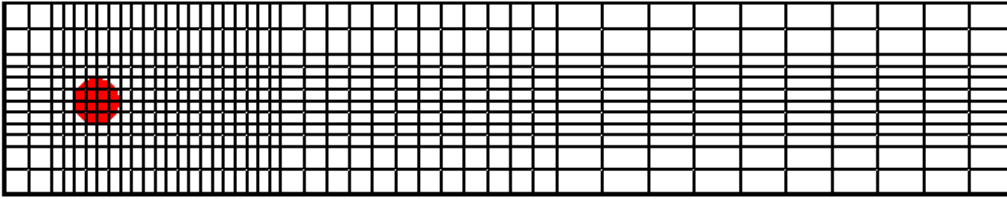## 3.1 Flow around a circular cylinder

We first consider the benchmark case of flow around a fixed circular cylinder in a channel as described in [20]. Fig. 2 shows a body-fitted mesh around the circular cylinder, as well as a cartesian grid for the FBM; the colored area shows the position of the cylinder. The shown (coarse) meshes are successively refined by connecting opposite midpoints. The channel height is $H = 0.41$ m, the cylinder diameter $D = 0.1$ m. The center 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 $\nu = \mu_f/\rho_f = 10^{-3} \, m^2/s$ and its density by $\rho_f = 1 \, kg/m^3$. 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 1 gives the parameters 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 in the CFD code FEATFLOW [24]. Compared to the body-fitted mesh, in the case of the fixed cartesian rectilinear mesh the cylinder shape is formed by the nodal points which cover the cylinder, instead 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 ≥ 4 on, we get an acceptable shape definition.

First of all, we test the steady case of Reynolds number Re = 20. Table 2 presents drag and lift coefficients by using the two different meshes: all results are convergent w.r.t. 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 for comparison. As can be seen, there is a good agreement between the presented method and the reference computation. For the simple mesh in Fig. 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 540672)! For increasing accuracy and reducing NEL, we tested an adapted mesh as shown in Fig. 3, which is refined near the wall surface of the cylinder. Table 3 shows the corresponding results for C_d and C_l. We can see that this special mesh on level 3 with only NEL= 2, 344 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 (see 'Conclusions').
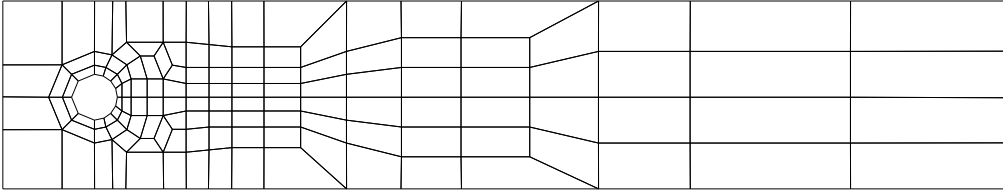
(a) body-fitted mesh (LEVEL = 1)

(b) cartesian mesh (LEVEL = 2) for multigrid FBM

Figure 2: Different coarse meshes adopted for flow around a fixed circular cylinder

Table 1: The parameters for sequentially refined meshes

LEVEL	body-fitted mesh				cartesian mesh				
	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	10608	2225	4336	2112	10784	95.493
4	8528	16848	8320	42016	8673	17120	8448	42688	97.482
5	33696	66976	33280	167232	34241	68032	33792	169856	99.472
6	133952	267072	133120	667264	136065	271232	135168	677632	99.721
7	534144	1066624	532480	2665728	542465	1083136	540672	2706944	99.814

Table 2: Drag and lift coefficients for case of Re = 20

LEVEL	body-fitted mesh		cartesian mesh	
	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
ref.	$C_d = 5.5795$		$C_l = 0.010618$	
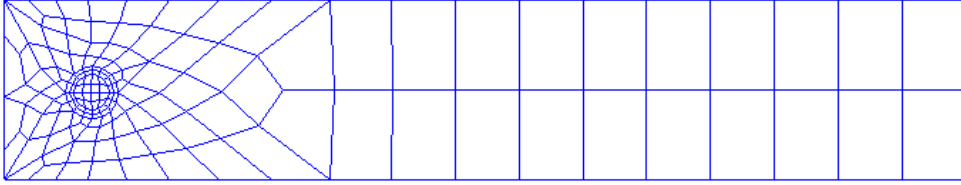
Figure 3: Special mesh for flow around a fixed cylinder (LEVEL = 1)

Table 3: Drag and lift coefficients on the adapted mesh for the case of Re = 20

LEVEL	NEL	C_d	C_l
3	2344	5.569	0.0002
4	9376	5.575	0.0014
5	37504	5.572	0.0081
6	150016	5.578	0.0102

Next we perform unsteady flow calculations with the higher Reynolds number Re = 100 using the same mesh as in Fig. 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 LEVEL = 7, are employed. Fig. 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 rectangular 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 rectilinear grid with LEVEL = 6 and on the finer LEVEL = 7, are more or less equal to those obtained from the reference calculation (see Fig. 4).
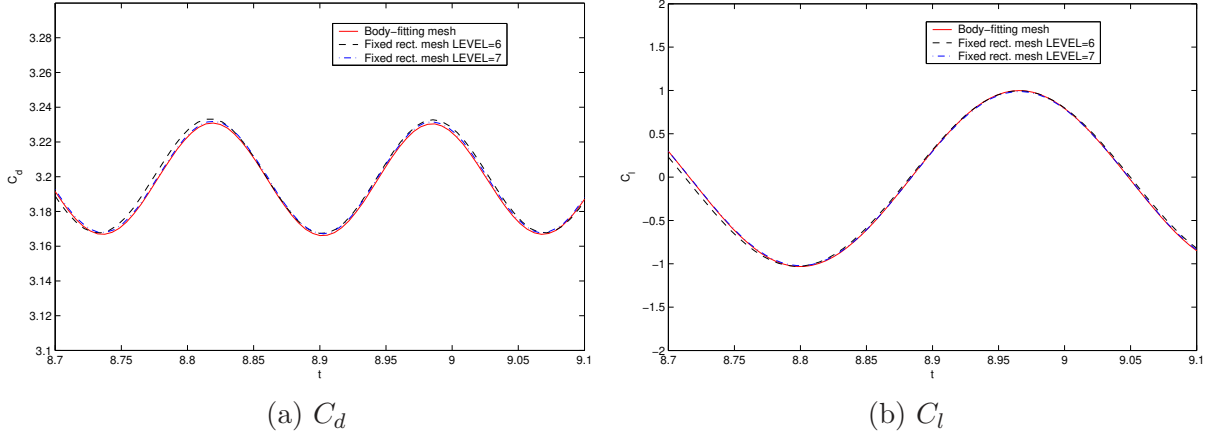
(a) C_d (b) C_l

Figure 4: Drag C_d and lift coefficient C_l of one period for Re = 100; solid line is for body-fitted mesh, dash line for fixed mesh LEVEL = 6; dot-dash line for fixed mesh LEVEL = 7

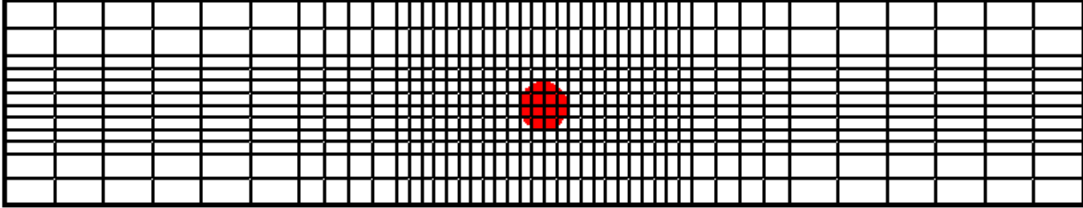## 3.2   An oscillating cylinder in a channel

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 computation 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 will be used for the fixed grid method. 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 [28] for details). The cylinder is moved with a prescribed velocity U_m and zero velocity conditions are imposed at the walls, inlet and outlet of the domain in the case of the fixed grid. In Fig. 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, two meshes on LEVEL = 6 and LEVEL = 7 are employed. Table 4 gives the parameters of the meshes in Fig. 5 with different numbers of refined levels.

Table 4: Grid characteristics for sequentially refined meshes

LEVEL	body-fitted mesh				cartesian mesh			
	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	10784
4	1888	3680	1792	9152	8673	17120	8448	42688
5	7360	14528	7168	36224	34241	68032	33792	169856
6	29056	57728	28672	144128	136065	271232	135168	677632
7	115456	230144	114688	574976	542465	1083136	540672	2706944
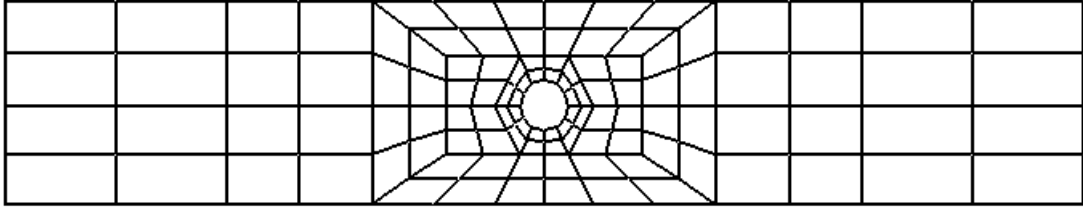
(a) body-fitted mesh (LEVEL = 2)


(b) cartesian rectilinear grid (LEVEL = 2)

Figure 5: Different 'coarse' meshes adopted for a moving cylinder in a channel

Fig. 6 illustrates the comparison of the drag coefficient C_d and the lift coefficient C_l between the results of the fictitious boundary method based on the channel mesh and the reference calculation based on the body-conformal mesh. The results calculated from LEVEL = 4 to LEVEL = 7 are all shown together (see Fig. 6 (a) and (b)). The corresponding coefficients C_d and C_l for one period between $t = 19.79$ to 23.79 are shown in Fig. 6 (c) and (d), the solid line represents the results of the reference calculation based on the body-conformal mesh at LEVEL = 7, while the dash line denotes the results obtained by the fictitious boundary method based on the channel mesh at LEVEL = 7. We can see that both FBM and reference results compare very well. The FBM results calculated by the presented fictitious boundary method agree very well with the reference results, although the FBM results exhibit small oscillations due to the non-aligned 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 very small.

4 Numerical schemes

4.1 Discretization in Space and Time

One of the common solution approaches for Eq. (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 [8] and [23]). Then, we obtain a sequence of generalized stationary Navier–Stokes

(a) C_d of FBM

(b) C_l of FBM

(c) one period of C_d

(d) one period of C_l
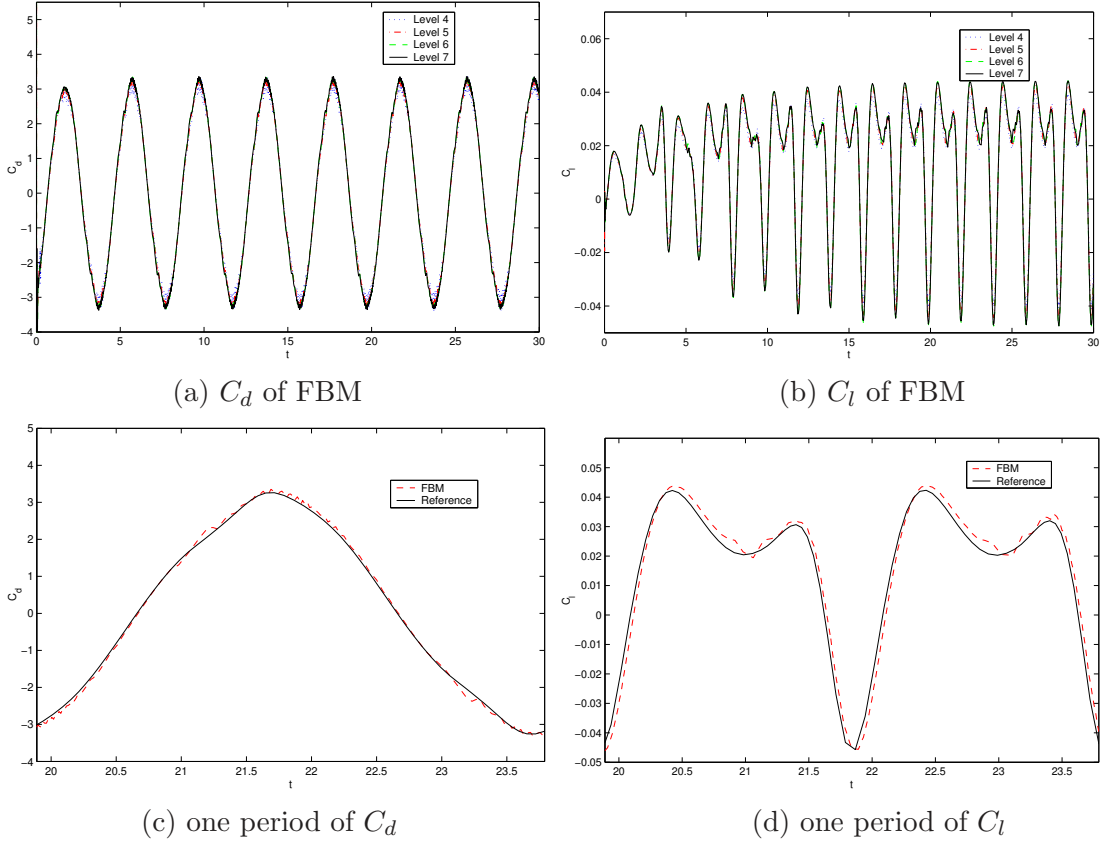
Figure 6: Comparison of C_d and C_l between FBM and reference

equations with prescribed boundary values for every time step. Here, we use the Fractional-step-θ-scheme [8] 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 (see [23]): 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 [23, 24].

4.2 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 [26]).

In this paper, based on the latter approach combined with multigrid methods, we adopt the discrete projection method (DPM [26]) 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 [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 [23] for the details).

4.3 Algorithmic Details

The whole algorithm of the multigrid 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 fictitious boundary conditions by using Eq. (10) (c) 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 Eqs. (10) (a) and (b) 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 Eq. (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*

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

where ϱ *is the maximum penetration distance to be allowed (maximal overlapping range).*

7. *By using the Newton-Euler equations of Eq. (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*

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

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

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

8. *Use the collision model of Eqs. (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*

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

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.*

4.4   Data Structures for Large Number of Particles

A challenging situation is the case with large numbers of rigid particles, for instance, the range of 100,000 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 6). To make it possible that the presented multigrid FBM method is able to simulate the particulate flows with a such 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. Because 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 FORTRAN) 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 analyze the cases of 10 to 100,000 particles falling down in a rectangular cavity filled with an incompressible Newtonian viscous fluid with and without these hierarchical techniques. In Table 5, "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 6 shows the typical CPU time needed (COMPAQ EV6, 666 MHz) for 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: '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 time reducing techniques are not used, the CPU time for the force calculations, as well as 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 1,000 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: Then, together with more advanced collision models and more efficient data structures, calculations with even 1,000,000 particles and more seem to be possible on modern PC.

Table 5: Parameters for the meshes in the test calculation

Level	NVT	NMT	NEL	NEQ
1	1406	27888	13824	69600
2	55777	111072	55296	277440
3	222145	443328	221184	1107840
4	886657	1771392	884736	4427520
5	3542785	7081728	3538944	17702400

Table 6: Typical CPU time for particle flow calculations (one time step) without (top) and with (bottom) the time reducing techniques

No. of Particles	= 10			= 100			= 1,000		
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			= 1,000			= 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**

5 Numerical Experiments

We present test configurations of benchmark character to evaluate and to validate the presented methodology. Flow with one rotating and moving particle is examined first to validate the calculated angular and translational velocities by the presented multigrid FBM. Then, cases with two and many particles are analyzed to check the quality of the collision model and of the efficiency of the FBM for the simulation of large numbers of particles. The time step is controlled automatically via an implicit estimation of the local truncation error of fluid velocities and pressure (see [23] for the details). In all simulations, we prescribe the geometrical and fluid quantities in dimensionless form.

5.1 One particle in a rotating circular container

We consider a particle with radius 1 centered in a circular container of radius 2 to validate the angular velocity, 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 mesh for the present calculation has 9,281 nodes and 9,216 elements. Fig. 7 shows the velocity distribution and the streamline contours at steady state with viscosity $\nu = 0.01$ and density $\rho = 1$. The angular velocity increases until it approaches the terminal angular speed which matches that of the wall of the container. Moreover, we present the terminal angular velocity of the particle and the time needed to reach the steady limit with different viscosities. When the viscosity ν becomes bigger, the particle can reach the steady state much faster.

| (a) velocity | (b) streamline |

viscosity ν	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
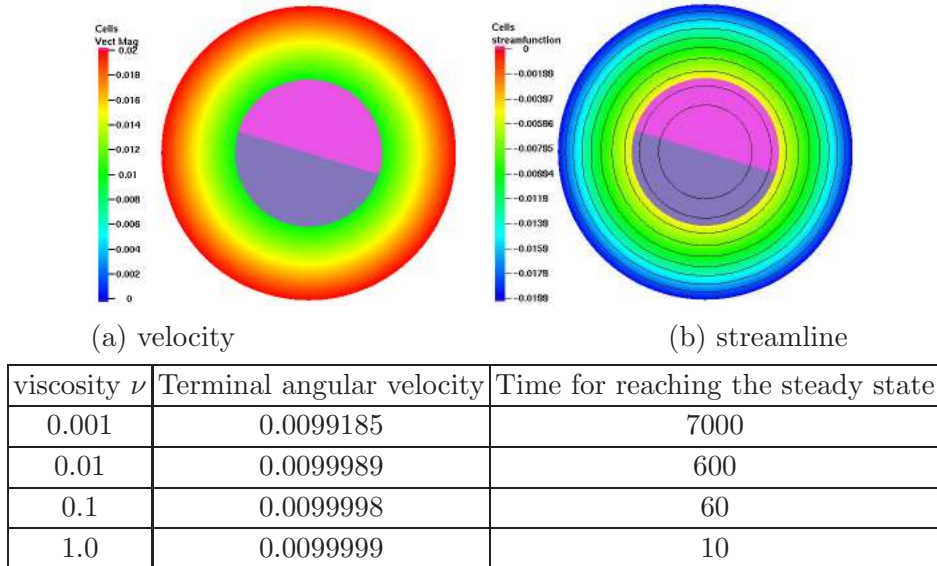
Figure 7: The motion of a particle in a rotating container and angular velocity at steady state

5.2 One particle in a shear flow

For further confirmation of the angular velocity, 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 viscosity is set to $\nu = 0.01$. The center 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 $U = -0.02$ so that, without any particle, the stationary flow would be a linear shear flow between the walls, with uniform vorticity $\Omega = 0.01$. The center of the particle is fixed, but it can rotate around its center. 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. Fig. 8 shows the velocity distribution at steady state, and Fig. 9 provides the evolution of the particle angular velocity in time for different radii R. Table 7 presents the terminal angular velocity of the particle and the time needed to reach the steady state with different radii of the particles. 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 this value due to less disturbance of the fluid field, and the particle can also reach the steady state much faster.

(a) $R \approx 0$ (b) $R = 1/20$ (c) $R = 1/10$

(d) $R = 1/5$ (e) $R = 2/5$ (f) $R = 1$
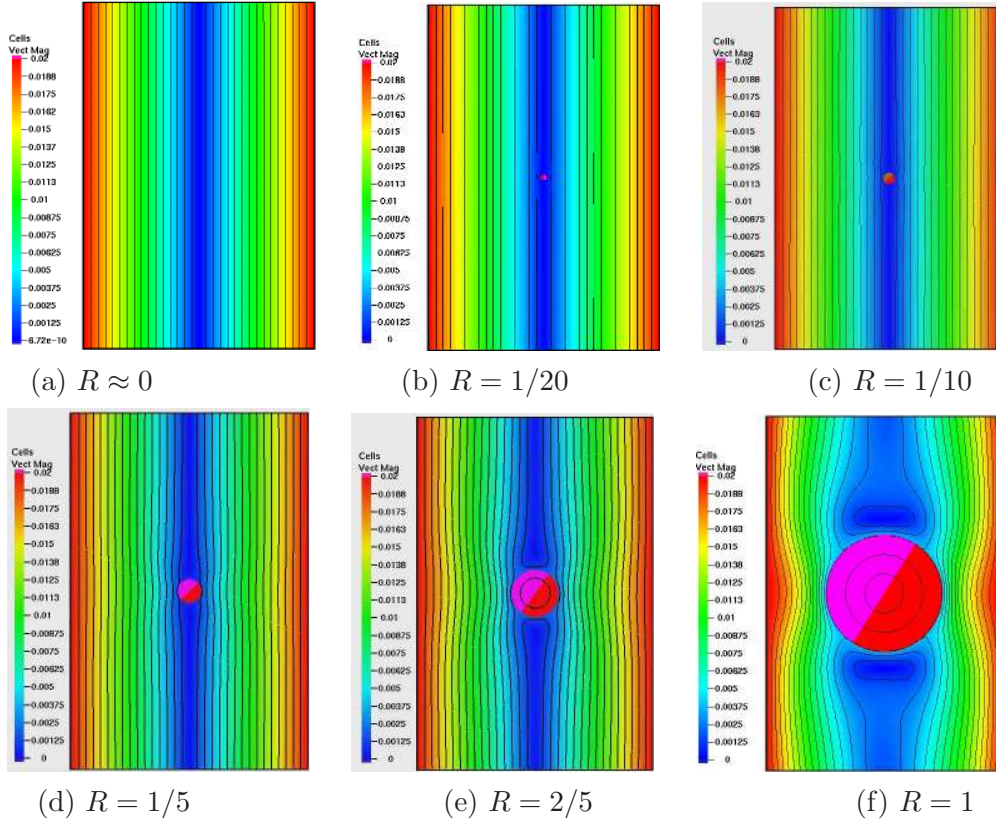
Figure 8: Norm of velocity distribution of a particle in shear flow
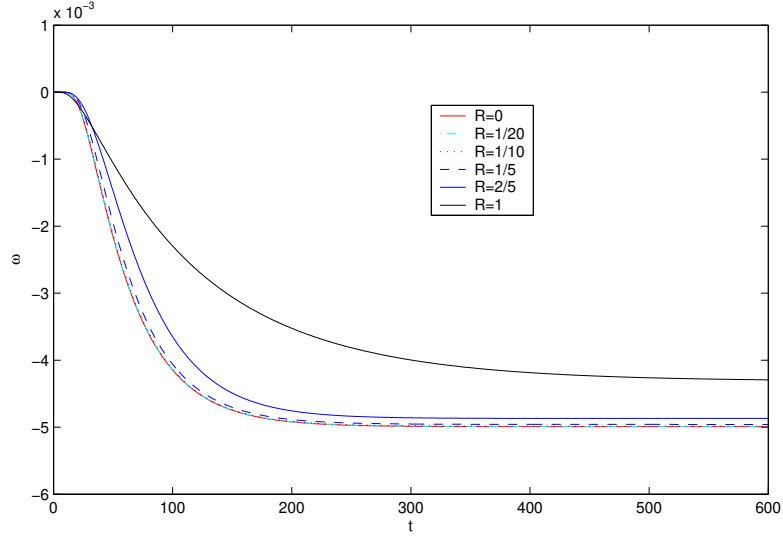
Figure 9: Angular velocity vs. time for a particle in shear flow

Table 7: Terminal angular velocity at steady state for a particle in 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

5.3 Flow around a moving airfoil

The rigid particles considered so far have been circular disks. The main goal of the following test problem is to show that the multigrid FBM can be 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 center of mass and is free to rotate due to hydrodynamical forces (see the description in [7, 8]). Fig. 10 shows the (fixed) coarse mesh which is locally adapted, the colored area is the position of the airfoil at rest; this coarse mesh is successively refined by connecting opposite midpoints. Table 8 gives the parameters for the refined meshes: "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 unknown. In this fixed mesh, the airfoil shape is formed by the nodal points which are inside of the airfoil, instead of a mesh line. "VEF" means the ratio of the effective airfoil area covered by the fixed mesh compared with the real airfoil area. We can see that from LEVEL ≥ 3 on, we get an acceptable shape approximation.

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

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

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 $\nu_f = 10^{-2}$. 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 $t \ge 0$. Here, the Reynolds number is approximately $Re \approx 101$ w.r.t the length of the airfoil and the maximum inflow speed [8].

In these simulations, we start the calculations with LEVEL $= 3$, until it reaches a stable periodical motion (here at $t = 42.05$), then we change to LEVEL $= 4$ to continue the calculation based on 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 Fig. 11. Fig. 12 presents the history of the angular velocity (ω) and the angle (θ) of the airfoil corresponding to the three different meshes. We observe that the results are in good agreement with those obtained by Glowinski, Joseph and coauthors [6, 8] and converge to a mesh independent solution after the flows reaches a periodic motion.
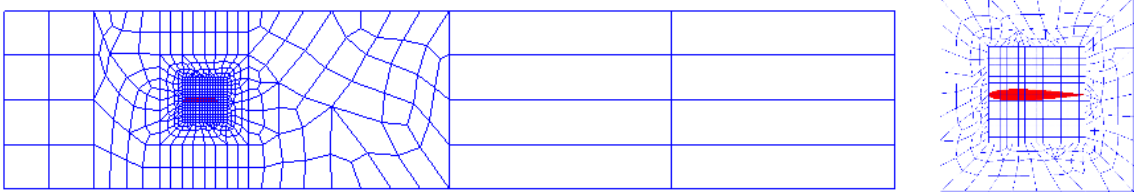
Figure 10: Coarse mesh for flow around a NACA0012 airfoil (left is the global mesh, right is a local zoom covering the airfoil at LEVEL $= 1$)

Table 8: Parameters for sequentially refined meshes

LEVEL	NVT	NMT	NEL	NEQ	VEF(%)
1	625	1220	596	3036	95.018
2	2441	4824	2384	12032	97.393
3	9649	19184	9536	47904	99.769
4	38369	76512	38144	191168	100.14
5	153025	305600	152576	763776	99.973

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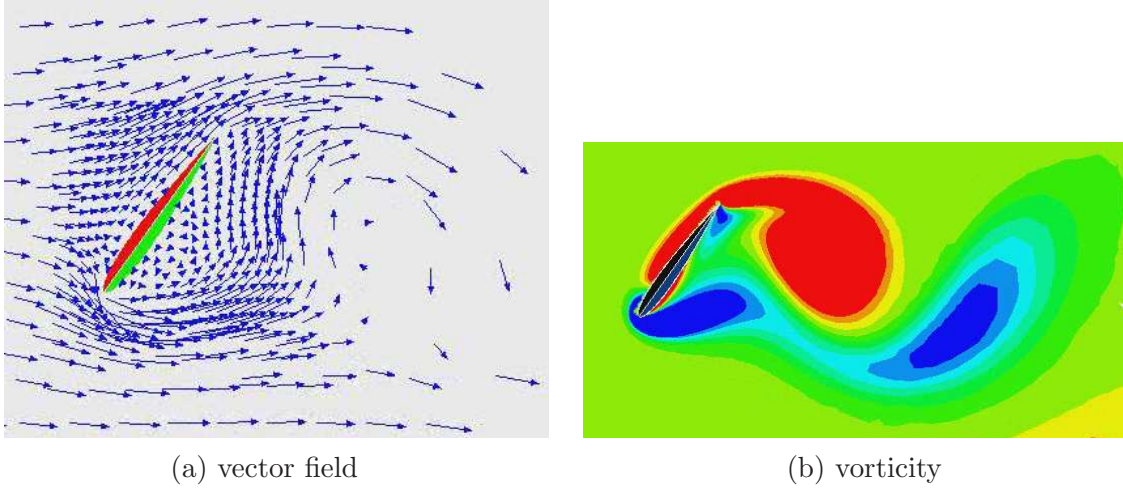
Figure 11: Snapshot for flow around a NACA0012 airfoil with $Re = 101$

(a) History of the angular velocity (rad/s)   (b) History of the angle (rad)
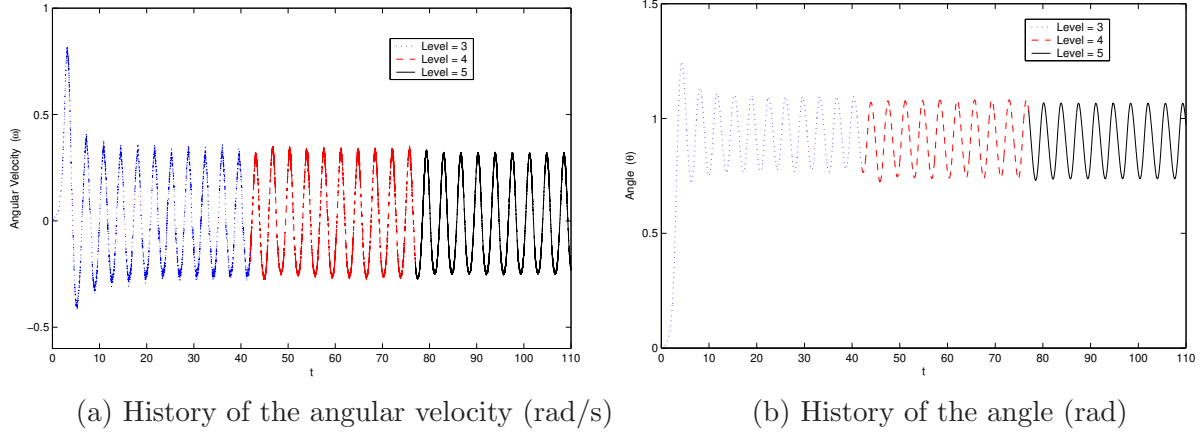
Figure 12: Measured data for flow around a NACA0012 airfoil with $Re = 101$, dotted line for LEVEL = 3, dashed line for LEVEL = 4, solid line for LEVEL = 5

5.4 Trajectory of a moving spinning ball

For verifying our multigrid FBM to simulate real particle motion with both angular velocity and translational velocities, we test a moving spinning ball. We begin the experiment by pushing and spinning the ball impulsively: 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 [11] 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 $(0, 20) \times (0, 4)$. 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.

Fig. 15 shows the measured data for the case $\rho_p = 10$ by using the two different meshes with Level $= 3$ and Level $= 4$ 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 velocity of the moving ball, (d) history of the v-component velocity of the moving ball, (e) history of 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 center 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 results are converged w.r.t. the mesh size and agree very well with the analytical results of Huang [11].
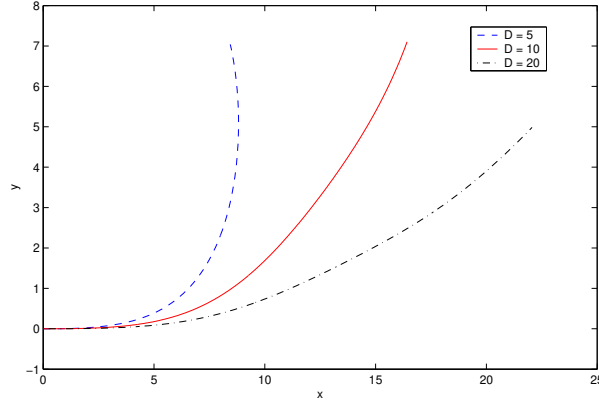
Figure 13: Trajectory for a moving ball with different densities at $t = 0.3$, dashed line for $\rho_p = 5$, solid line for $\rho_p = 10$, dot-dashed line for $\rho_p = 20$ by using the Level $= 4$ mesh
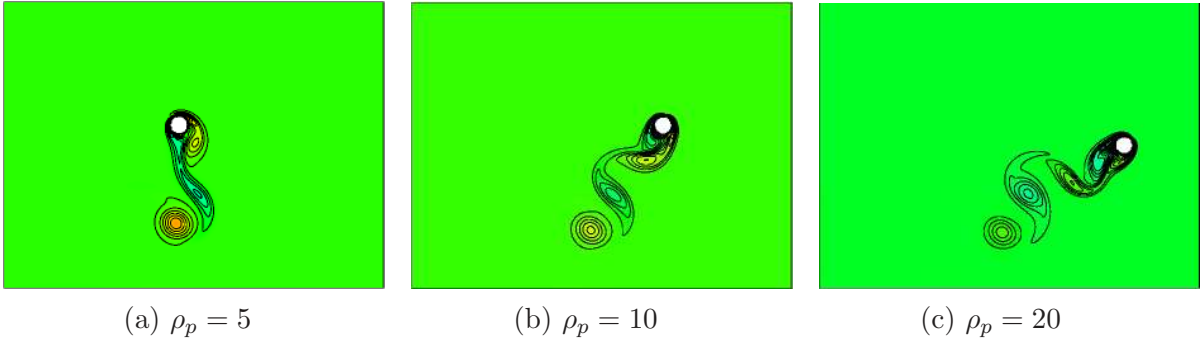


(a) $\rho_p = 5$ (b) $\rho_p = 10$ (c) $\rho_p = 20$

Figure 14: Contour plots of vorticity at $t = 0.3$ for a moving spinning ball

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 $\rho_p = 20$. Fig. 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. Fig. 14 finally shows the contour plot of vorticity at $t = 0.3$ for the three different ball densities $\rho_p = 5$, 10 and 20.
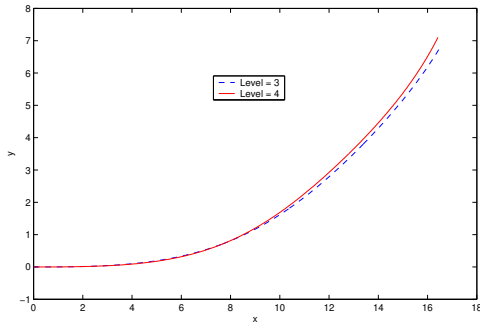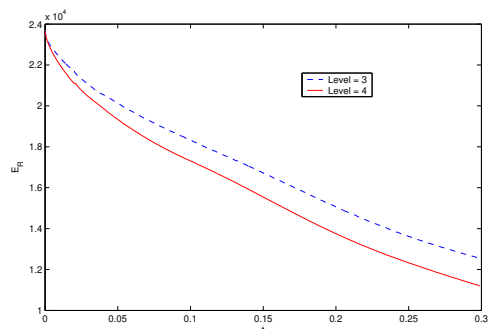
(a) Trajectory at $t = 0.3$
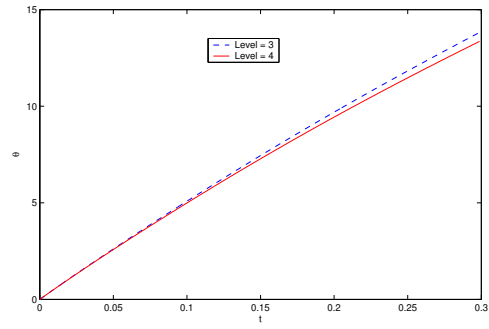
(b) Angle θ with time
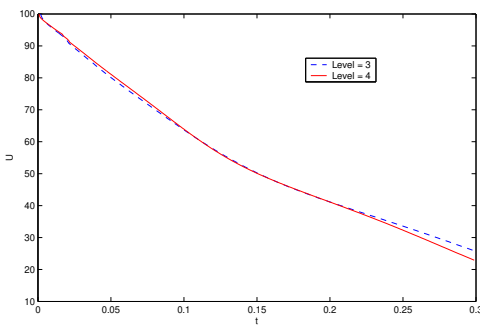
(c) u-component with time
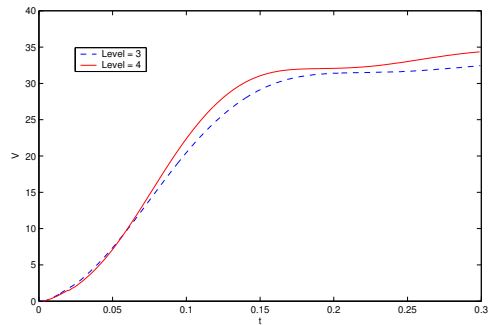
(d) v-component with time
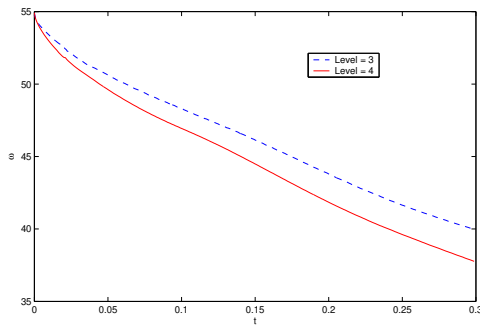
(e) Angular velocity ω with time
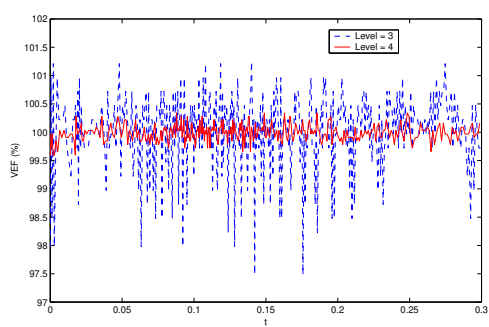
(f) Effect volume coeff. with time
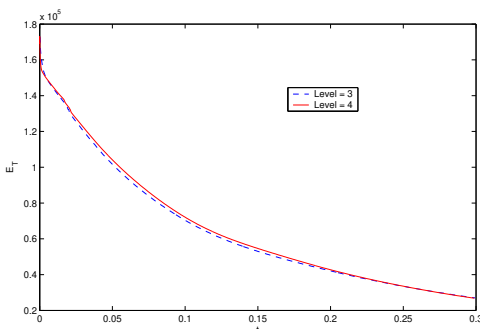
(g) Translational kinetic energy with time

(h) Rotational kinetic energy with time

Figure 15: Calculated data for a moving ball for $\rho_p = 10$

5.5   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 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 $\nu = 0.1$. 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.

Fig. 16 shows the velocity field computed with $\Delta h = 1/96$ on Level $= 4$. We can see that the disk quickly reaches a uniform falling velocity until it hits the bottom of the channel.
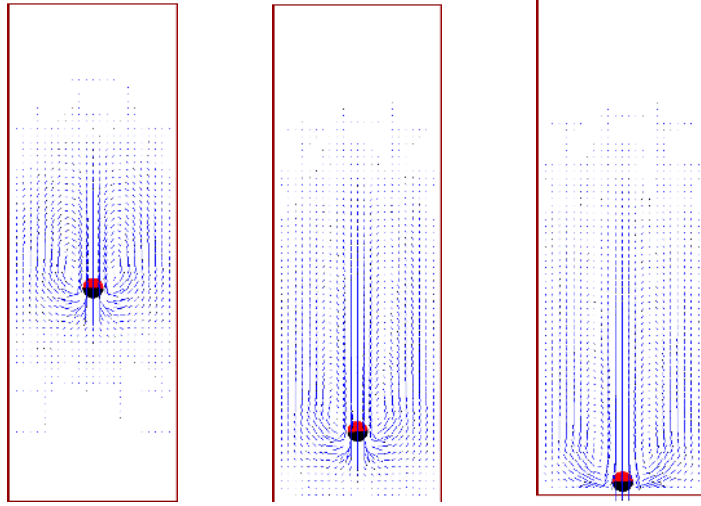

Figure 16: One circular disk falling down in a fluid: (a) $t = 0.34$, (b) $t = 0.66$, (c) $t = 0.79$

In Fig. 17, some of the quantities are plotted in time: these are the y-coordinate of the disk center, v-component of the disk translational velocity, Reynolds number for the disk defined by $Re = \sqrt{u^2 + v^2} \cdot d \cdot \rho_p / \nu$, 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 center 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 accuracy on Level $= 4$ is better 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 [8]. We can see that the results computed on the two different mesh sizes are essentially the same except the case of the rotational kinetic energy due to its very small values ($< 10^{-3}$) which is easily disturbed by numerical errors.
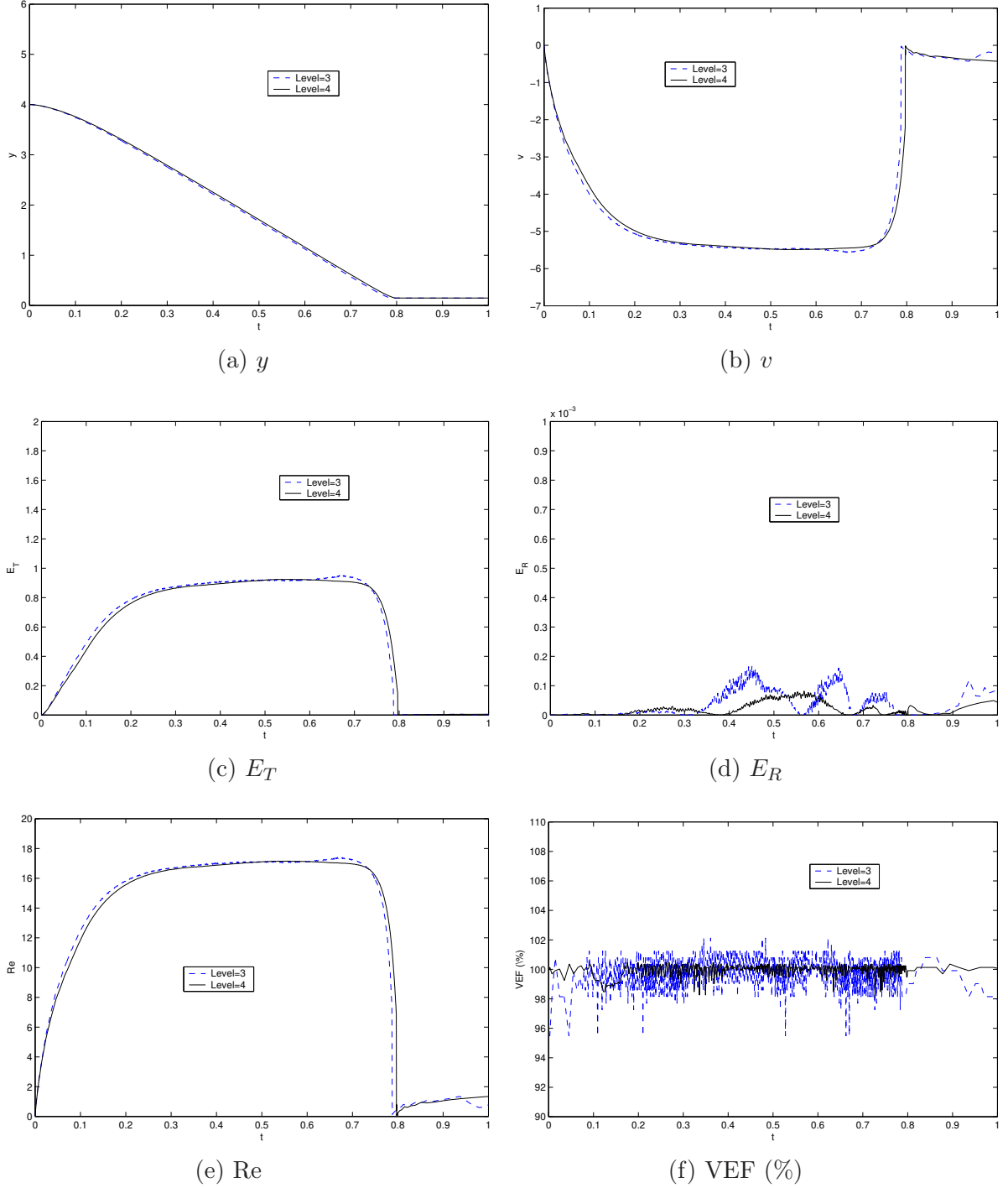
Figure 17: One circular disk falling down in a fluid ($\rho_p = 1.25$): time histories of the y-coordinate (a) of the particle center, v-component (b) of the particle translational velocity, Reynolds number for the disk (c), and translational kinetic energy (d), rotational kinetic energy (e) as well as the effective area (f) of the disk covered by the used equidistant mesh compared with the real area of the disk, dashed line for $\Delta h = 1/48$ (Level = 3), solid line for $\Delta h = 1/96$ (Level = 4)

5.6 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 w.r.t. two different mesh sizes and regarding the results in [8]. 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 $\nu = 0.01$. The simulation is started at $t = 0$ by dropping two particles at the center of the channel at a height of 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 computed solutions w.r.t. the mesh size.
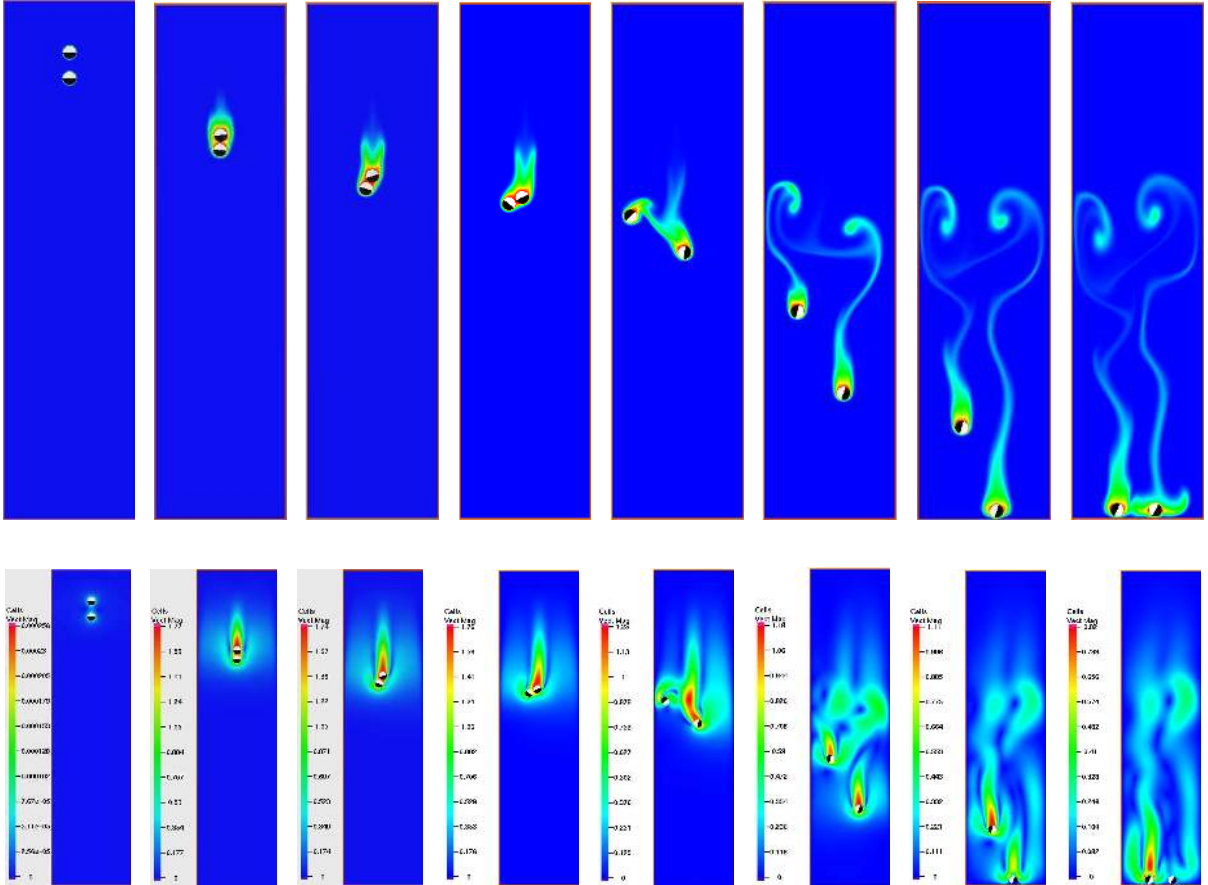

Figure 18: Sedimentation of two circular particles: (a) $t = 0.00$, (b) $t = 1.13$, (c) $t = 1.53$, (d) $t = 1.73$, (e) $t = 2.44$, (f) $t = 4.49$, (g) $t = 6.23$, (h) $t = 7.57$ (from left to right)

It is well known that when two particles are dropped close to each other, they interact by undergoing "drafting, kissing and tumbling" [5]. Fig. 18 shows the visualization of the characteristic drafting, kissing and tumbling behavior (top) of two falling particles, computed with $\Delta h = 1/96$ on Level = 4, providing also the corresponding velocity distribution (bottom).

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 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 $t = 7.57$.
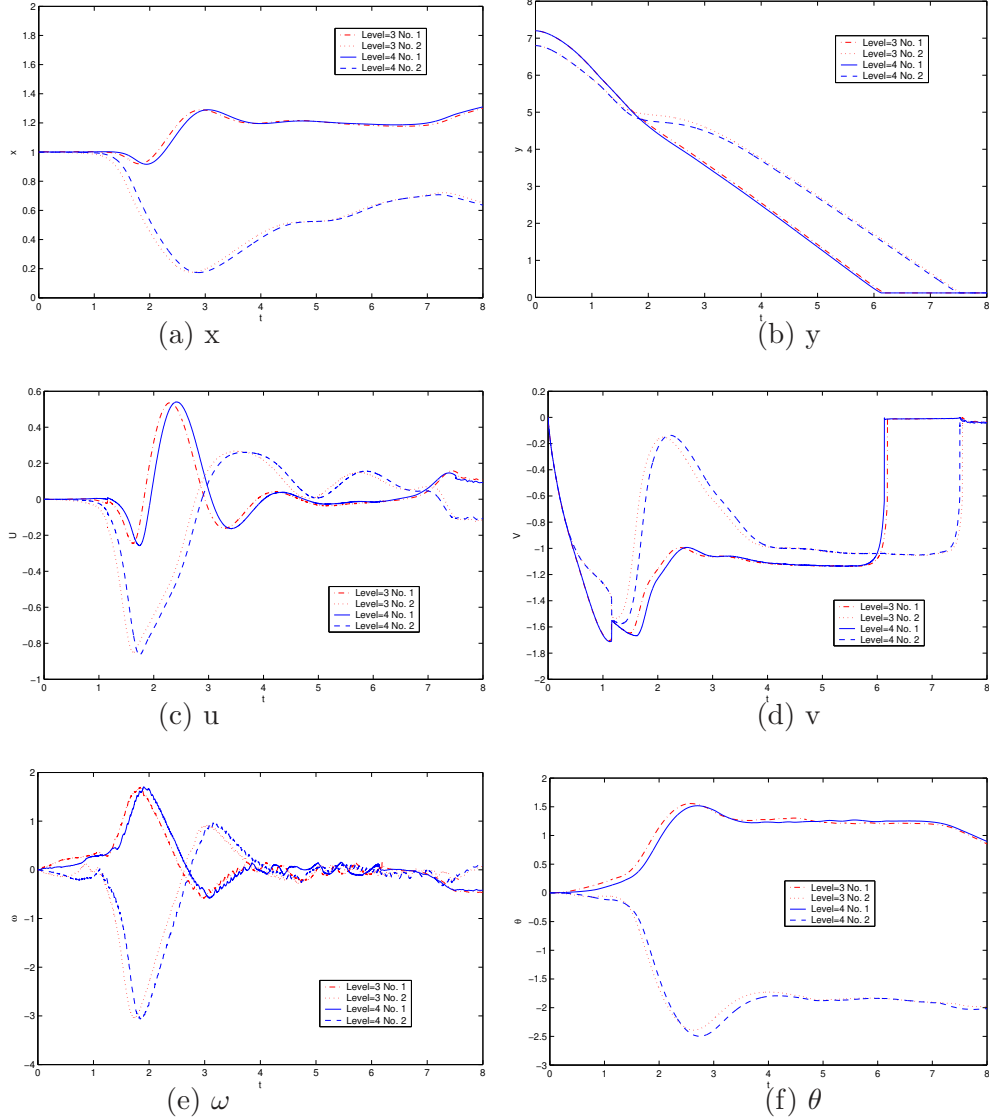

Figure 19: Sedimentation of two circular particles: time histories of x-coordinates (a), y-coordinates (b) of the particle center, u-component (c), v-component (d) of the particle translational velocity, and of the particle angular velocity (e), and the particle angle (f) of the particle, obtained for 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)

In Fig. 19 and Fig. 20, several quantities are plotted: these are the time histories of the x-coordinates and y-coordinates of the two particle centers, u-component and v-component of the particle translational velocities, the particle angular velocities, and the particle angles, as well as 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, 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.
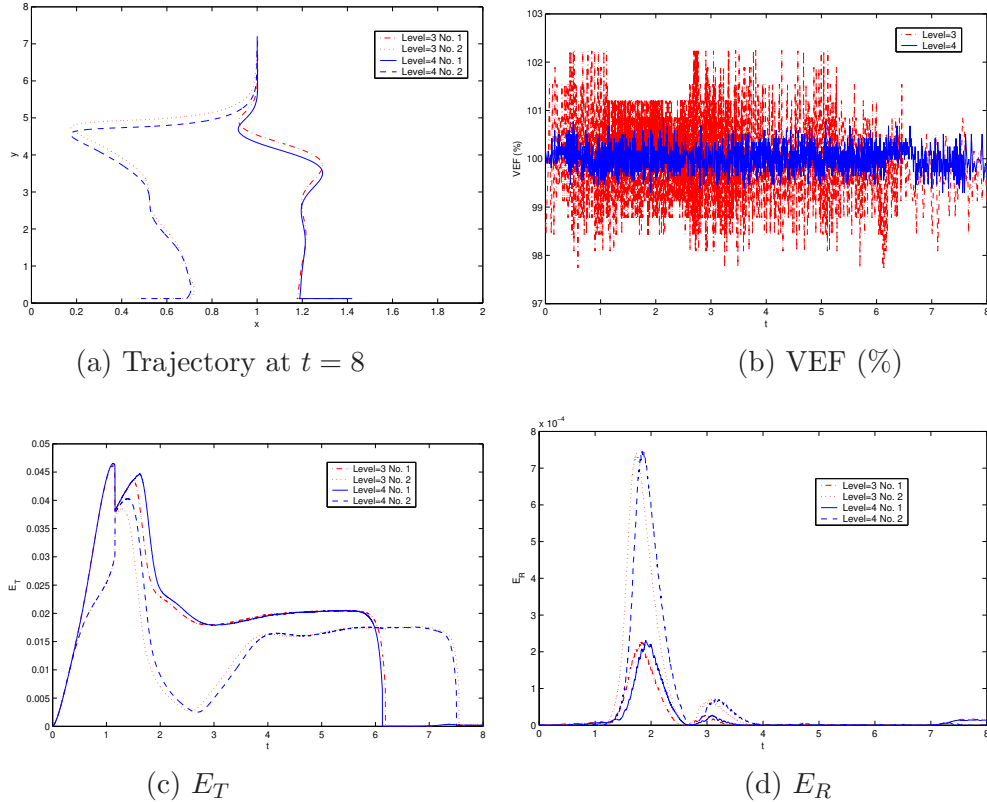

(a) Trajectory at $t = 8$ (b) VEF (%)

(c) E_T (d) E_R

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 the translational (c) as well as of the rotational kinetic energy (d)

The two particles are kissing each other at $t = 1.13$. After kissing, they tumble: a manifestation of the instability of a falling long body aligned with the stream. The histories of the particle angles and the translational and rotational kinetic energies are presented for both mesh sizes. 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 [8, 21, 19, 3]. However, there is still need for further rigorous benchmarks for multiple particles.

5.7 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 particle 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 [8]), from simulations or experiments, so that we can only examine the qualitative behavior.

The specific problem in this subsection is the sedimentation of 790 circular particles with different diameters settling 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 such that in each row the diameter of the particles is the same. The number of particles at the top row 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 Fig. 21 (a): particles and 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 $\nu = 10^{-2}$. The parameters of the mesh have been listed in Table 5. The parameter ϵ_P in the collision model has been taken equal to 5×10^{-7}, and $\epsilon_W = \epsilon_P/2$, $\epsilon'_P = \epsilon_P$, $\epsilon'_W = \epsilon_W$. The evolution of the 790 particles is shown in Figs. 21 and 22. A Rayleigh-Taylor instability develops in the advancing front between $t = 0.52$ and $t = 1.60$. 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 $t = 2.68$, 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.

5.8 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 Fig. 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 non-dimensional 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 $\epsilon_W = \epsilon_P/2$, $\epsilon'_P = \epsilon_P$, $\epsilon'_W = \epsilon_W$. The evolution of the 3600 circular particles is shown in Figs. 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: 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.
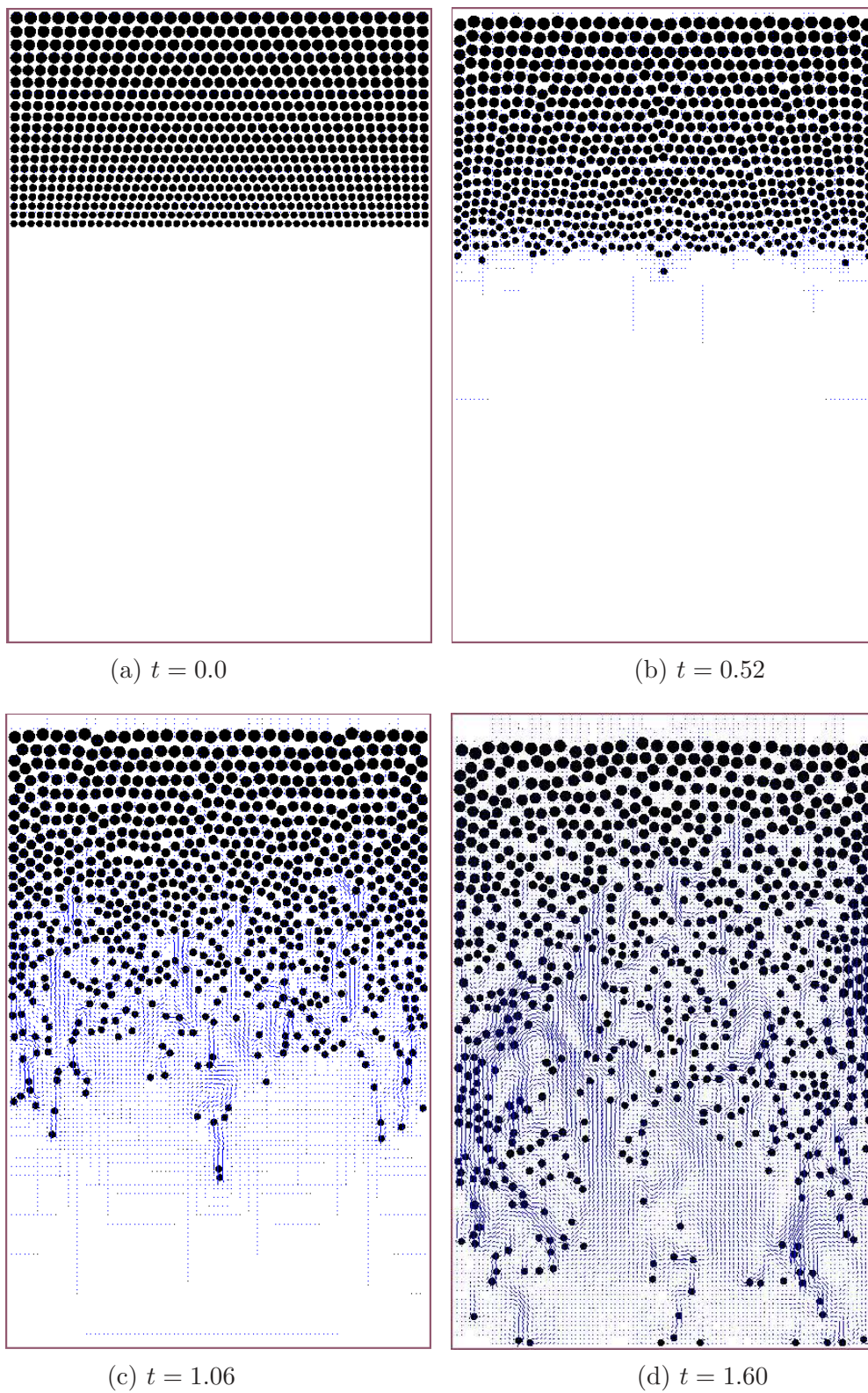
(a) $t = 0.0$

(b) $t = 0.52$

(c) $t = 1.06$

(d) $t = 1.60$

Figure 21: Velocity field for 790 particles with various diameters falling down in a cavity

(e) $t = 2.14$

(f) $t = 2.68$

(g) $t = 4.02$

(h) $t = 6.19$
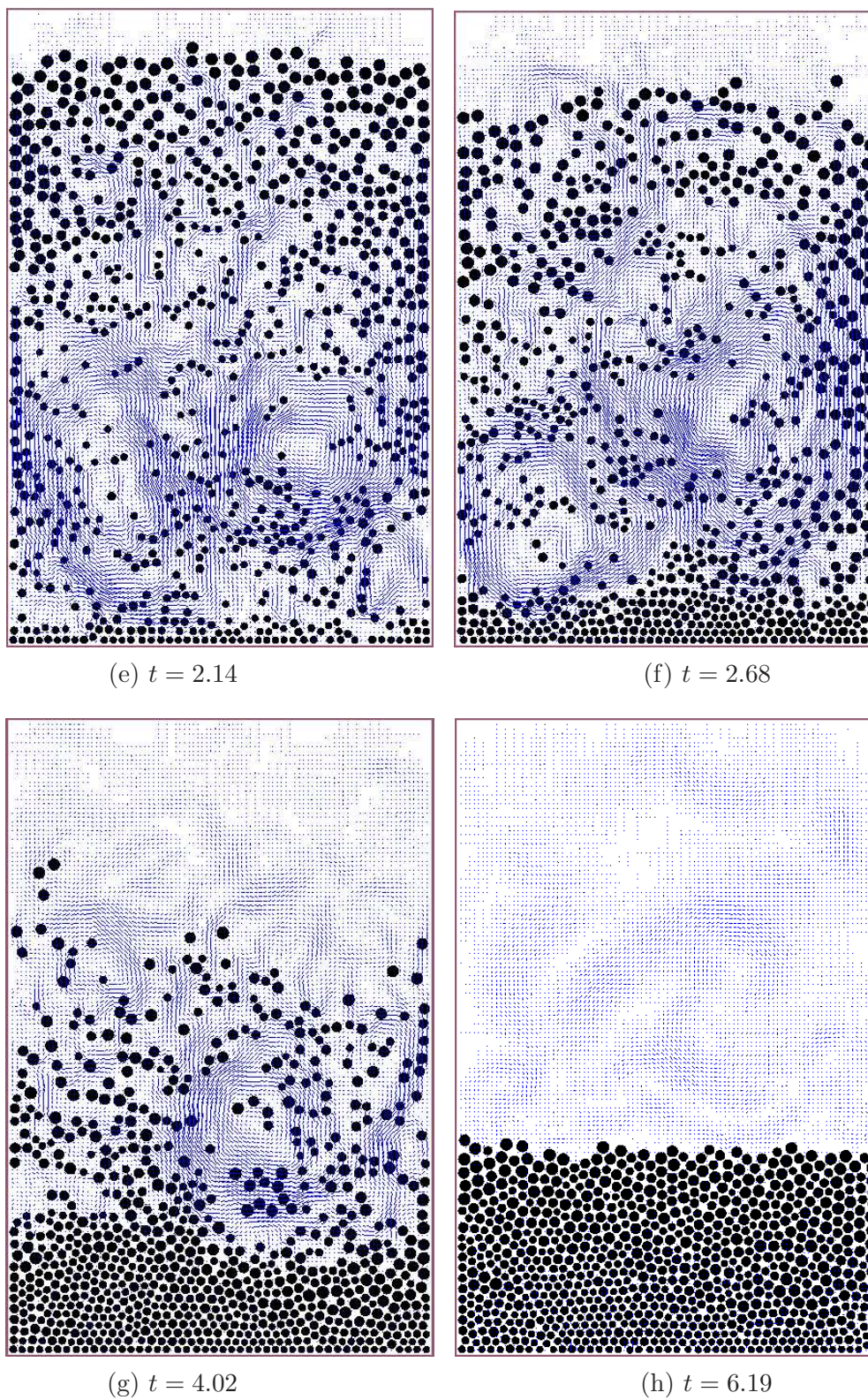
Figure 22: Velocity field for 790 particles with various diameters falling down in a cavity (cont.)

(a) $t = 0.0$

(b) $t = 0.52$

(c) $t = 1.06$

(d) $t = 1.60$
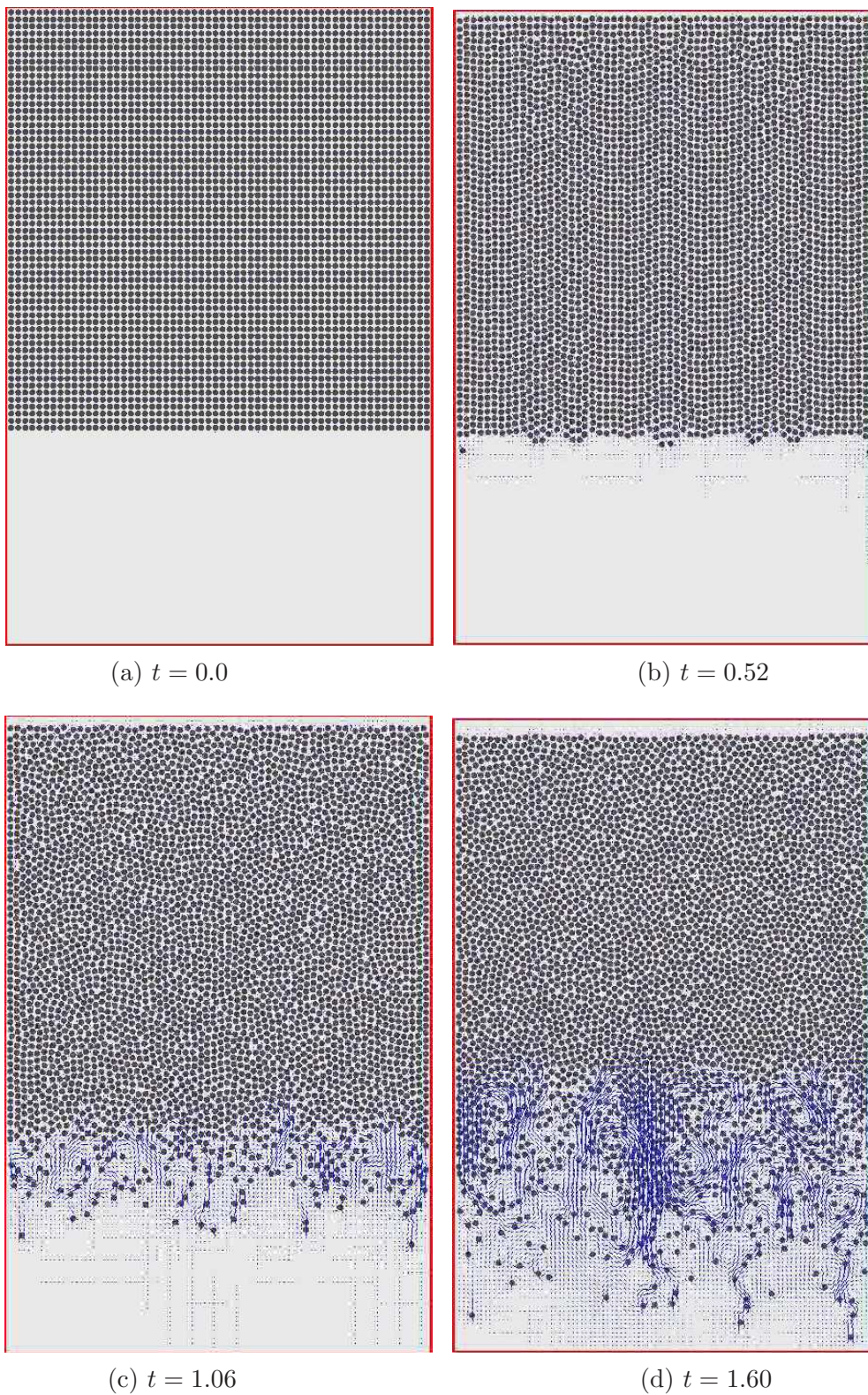
Figure 23: Velocity fields for 3600 particles falling down in a cavity

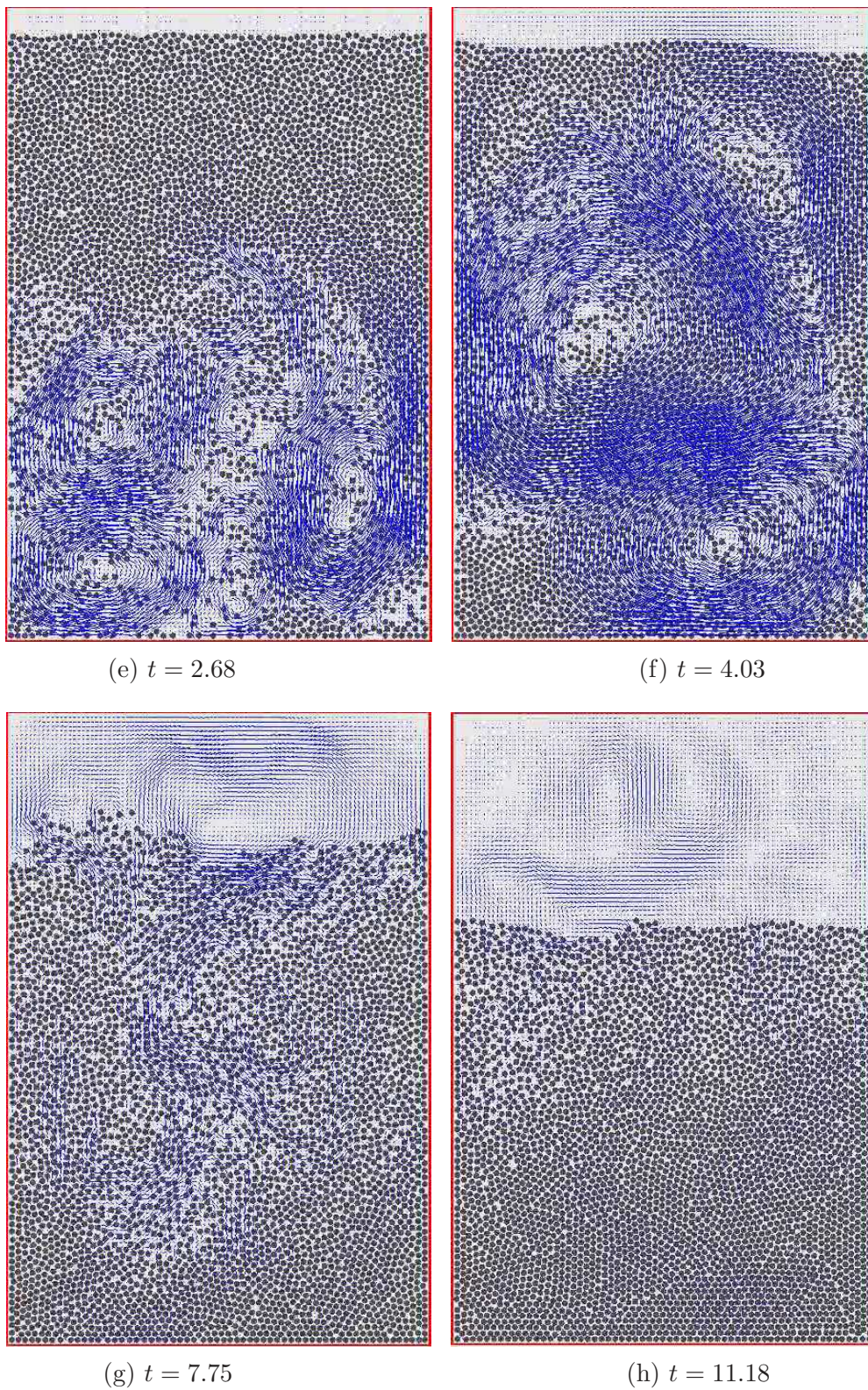(e) $t = 2.68$

(f) $t = 4.03$

(g) $t = 7.75$

(h) $t = 11.18$

Figure 24: Velocity fields for 3600 particles falling down in a cavity (cont.)

6 Conclusions

We have presented the multigrid fictitious boundary method (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 analyzed methodology in the 3D solvers of our CFD package FeatFlow [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 the paper. The proposed volume-based integration for the calculations 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 instance [8, 19], is not yet clear, particularly w.r.t. 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 these 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, non-Newtonian and viscoelastic fluids ("chaining" instead of "kissing, drafting, tumbling", see [8]) have to be tackled in 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 fictitious boundary method 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 is important 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 the following Fig. 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 recent papers by Liao et al. [2, 15] 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 6). Therefore, we plan to present the details of this special adaptive approach in a forthcoming paper.
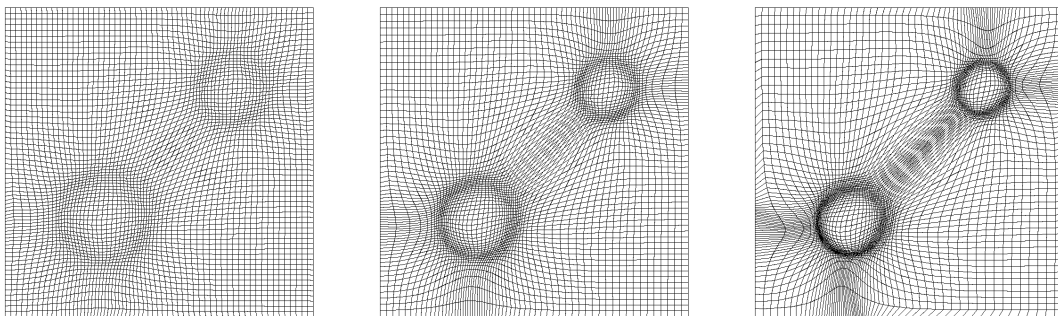

Figure 25: Deformed meshes with different 'grid deformation' parameters

References

[1] Asakura, K., Asari, T. and Nakajima, I.: Simulation of solid-liquid flows in a vertical pipe by a collision model, Powder Technology, **94**, (1997), 201-206

[2] Bochev, P., Liao, G., dela Pena, G.: Analysis and computation of adaptive moving grids by deformation, Numer. Meth. Part. Diff. Eq., **12**, (1996), 489

[3] Diaz-Goano, C., Minev, P. and Nandakumar, K.: A Lagrange multiplier/fictitious domain approach to particulate flows, in: Margenov, W., Yalamov (Eds), Lecture Notes in Computer Science, Vol. 2179, Springer, (2001), 409-422

[4] Duchanoy, C. and Jongen, T.R.G.: Efficient simulation of liquid-solid flows with high solids fraction in complex geometries. Compters and Fluids **32**, (2003), 1453

[5] Fortes, A., Joseph, D.D. and Lundgren, T.: Nonlinear mechanics of fluidization of beds of spherical particles, J. Fluid Mech., **177**, (1987), 497-483

[6] Glowinski, R., Pan, T.W., Hesla, T.I. and Joseph, D.D.: A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow, **25**, (1999), 755-794

[7] Glowinski, R., Pan, T.W., Hesla, T.I., Joseph, D.D. and Periaux, J.: A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow, J. Comput. Phy., **169**, (2001), 363-426

[8] Glowinski, R.: Numerical methods for fluids (Part 3), Handbook of numerical analysis, Volume IX, Ciarlet, P.G and Lions, J.L., Editors, North-Holland, (2003)

[9] Hu, H.H., Joseph, D.D. and Crochet, M.J.: Direct simulation of fluid particle motions, Theor. Comp. Fluid Dyn. 3, (1992), 285-306

[10] Hu, H.H.: Direct simulation of flows of solid-liquid mixtures, Int. J. of Multiphase Flow. **22**, (1996), 335-352

[11] Huang, J. Y.: Trajectory of a moving curveball in viscid flow, Proceedings of the International Conference on Dynamical Systems and Differential equations, Atlanta, USA, May 18-21, (2000), 1-7

[12] John, V.: Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations. Int. J. Num. Meth. Fluids **40**, (2002), 775

[13] Johnson, A.A. and Tezduyar, T.E.: Simulation of multiple spheres falling in a liquid-filled tube, Comput. Methods Appl. Mech. Engrg., **134**, (1996), 351-373

[14] Leal, L.G.: Laminar flow and convective transport processes, Scaling Principles and Asymptotic Analysis, Butterworth-Heinemann, Newton, Massachusetts

[15] Liao, G., Fleitas, D., Jiang, B.: Adaptive grid generation based on least-squares finite element method, Technical Report, Dpt. of Math., University of Texas, (2004)

[16] Maury, B.: Characteristics ALE method for the unsteady 3D Navier-Stokes equations with a free surface, Int. J. of Comput. Fluid Dynamics, J. Comput. Phy., **6**, (1996), 175-188

[17] Maury, B.: A many-body lubrication model, C. R. Acad. Sci. Paris. T. 325, Série 1, (1997), 1053-1058

[18] Maury, B.: Direct Simulations of 2D fluid-particle flows in biperiodic domains, J. Comput. Phy., **156**, (1999), 325-351

[19] Patankar, N.A., Singh, P., Joseph, D.D., Glowinski, R. and Pan, T.W.: A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow, **26**, (2000), 1509-1524

[20] Schäfer, M. and Turek, S.: Benchmark computations of laminar flow around cylinder. in E.H. Hirschel (editor) Flow Simulation with High-Performance Computers II. Volume 52 of Notes on Numerical Fluid Mechanics, Vieweg (1996), 547

[21] Singh, P., Hesla, T.I. and Joseph, D.D.,: Distributed Lagrange multiplier method for particulate flows with collisions. Int. J. Multiphase Flow, **29**, (2003), 495-509

[22] Sommerfeld, M.: Validation of a stochastic Lagrangian modelling approach for intre-particle collisions in homogeneous isotropic turbulence, Int. J. Multiphase Flow, **27**, (2001), 1829-1858

[23] Turek, S.: Efficient Solvers for Incompressible Flow Problems. Springer Verlag, Berlin-Heidelberg-New York, (1999)

[24] Turek, S.: **FEATFLOW** *Finite element software for the incompressible Navier–Stokes equations: User Manual, Release 1.1*, University of Heidelberg, (1998)

[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, Int. J. Numer. Meth. Fluids, 22, (1996), 987-1011

[26] Turek, S.: On discrete projection methods for the incompressible Navier–Stokes equations: An algorithmical approach, Comput. Methods Appl. Mech. Engrg., 143, (1997), 271 - 288

[27] Turek, S., Wan, D.C. and Rivkind, L.S.: 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, (2003), 37-68

[28] Wan, D.C., Turek, S. and Rivkind, L.S.: An efficient multigrid FEM solution technique for incompressible flow with moving rigid bodies, Numerical Mathematics and Advanced Applications, ENUMATH 2003, Springer, (2004), 844-853