

# An Efficient Multigrid-FEM Method for the Simulation of Liquid-Solid Two Phase Flows

Decheng Wan and Stefan Turek

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

---

## Abstract

An efficient multigrid-FEM method for the detailed simulation of solid-liquid two phase flows with large number of moving particles is presented. An explicit fictitious boundary method based on a FEM background grid which covers the whole computational domain and can be chosen independently from the particles of arbitrary shape, size and number is used to deal with the interactions between the fluid and the particles. Since the presented method treats the fluid part, the calculation of forces and the movement of the particles in a subsequent manner, it is potentially powerful to efficiently simulate real particulate flows with huge number of particles. The presented method is first validated using a series of simple test cases, and then as an illustration, simulations of three big particles plunging into 2000 small particles, and of sedimentation of 10,000 particles in a cavity are presented.

*Key words:* Liquid-Solid Flows, Multigrid FEM, Fictitious Boundary Method

---

## 1 Introduction

Solid-liquid two phase flows are ubiquitous in chemical, pharmaceutical and food industries as well as geophysical environments, including debris flows, slurries, mining and milling operations, sedimentation columns and fluidized beds, lubricated transport, and hydraulic fracturing.

Direct numerical simulation of solid-liquid two phase flows is a difficult task since the domain occupied by the fluid is irregular and changes with motion of the particles. Also the particles are advected by the fluid and exert forces at the fluid, so the body-liquid interaction requires calculation of the fluid stress at the fluid-solid interface, 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 give further complexity to the problem.

There are two separate approaches having been developed to solve such problem. The first is a generalized ALE standard Galerkin finite element method [1–6] in which both the fluid and particle equations of motion are incorporated into a single coupled variational equation. Both the fluid and particle velocities appear as primitive unknowns. The hydrodynamic forces and torques on the particles are eliminated in the formulation, so need not to be computed as separate quantities. The computation is performed on an unstructured body-fitted grid, and an arbitrary Lagrangian-Eulerian (ALE) moving mesh technique is adopted to deal with the motion of the particles. The nodes on the particle surface move with the particle, while the nodes in the interior of the fluid are computed using Laplace’s equation to guarantee a smoothly varying distribution of nodes. At each time step, the grid is updated according to the motion of the particles and checked for element degeneration. If unacceptable element distortion is detected, a new finite element grid is generated and the flow fields are projected from the old grid to the new grid. In this scheme, the positions of the particles and grid nodes are updated explicitly, while the velocities of the fluid and the solid particles are determined implicitly.

The second approach is based on the principle of embedded or fictitious domains. The idea is to embed an irregular computational domain into a larger, simpler domain, and to specify simple boundary conditions on its boundary. The fluid flow is computed as if the space occupied by the particles were filled with fluid. The no-slip boundary condition on the particle boundaries is enforced as a constraint. This allows a fixed grid to be used, eliminating the need for remeshing, a definite advantage in parallel implementations. There are several ways to apply this principle to solve the problem of fluid flow around moving obstacles. Glowinski, Joseph and coauthors [7–9] developed a distributed Lagrange multiplier (DLM)/fictitious domain method. In the DLM method, usually referred to as body-force-DLM method, the entire fluid-particle domain is assumed to be a fluid and then to constrain the particle domain to move with a rigid motion. The fluid-particle motion is treated implicitly using a combined weak formulation in which the mutual forces cancel. Patankar, Joseph, Glowinski and coauthors [10] presented a variant of the DLM method, namely stress-DLM method, in which the rigid motion is imposed by constraining the deformation-rate tensor within the particle domain to be zero. This eliminates the translational and angular velocities of the particles as variables from the coupled system of equations. It recognizes that the rigidity constrain results in a stress field inside a rigid solid just as there is pressure in an incompressible fluid.

Recent computational approaches to solid-liquid flows, possibly inspired by molecular dynamics, are cellular automata and the lattice Boltzmann method (LBM) [11–13]. In LBM, simplified kinetic models, which incorporate the essential physics of the microscopic and mesoscopic equations, are constructed. These models can handle huge numbers of particles. However, they replace

the equations of motion with computer rules and do not deal with stagnation and separation points, wakes, turning couples, drafting, kissing and tumbling, etc. The interesting results produced by these methods are not yet sufficiently reliable to be used in engineering practice.

In aforementioned DLM methods, they are often referred to as an implicit fictitious boundary approach since there is no need to directly calculate the hydrodynamic forces exerted on the particles. The obvious advantage of the implicit fictitious boundary approach is that the computational time for calculation of forces exerted on particles can be saved. However, the implicit coupling of fluid-solid momentum equations slows down the solution procedure, since it requires the solution of large systems of the linear and nonlinear algebraic equations for the coupled variables of fluid and solid. For the case involving a large number of particles (e.g.  $\geq 10,000$ ), these systems can be extremely large. In contrast to the implicit fictitious boundary approach, an explicit fictitious boundary approach is to solve fluid equations and solid equations separately. The forces exerted on particles are calculated in a very efficient way. The computational costs are practically independent of the number of particles presenting in the computational domain. The explicit fictitious boundary approach can be expected to be more powerful than the implicit one in simulating real particulate flows with large number of particles. Duchanoy and Jongen [14] developed a finite volume method based explicit fictitious boundary method to efficiently simulate the food processing in tubular heat exchangers. Turek, Wan and Rivkind [15] proposed a multigrid FEM based explicit fictitious boundary method (FBM). The method is based on an unstructured FEM background grid. 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 same fixed grid is also used to represent the location of the solid particles by imposing the velocities on the nodes covered by the particles at any time. The new positions and the new velocities of the particles are updated using Newton's law so that there is no need to remesh the domain. 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. 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. It 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.

In this paper, we adopt the multigrid FEM fictitious boundary method to simulate solid-liquid two phase flows with huge number of moving particles in fluid. The accuracy and efficiency of the presented method are first validated using a series of simple test cases, and then as an illustration, numerical results of three big particles plunging into 2000 small particles, and sedimentation of 10,000 particles in a cavity are presented.

## 2 Governing Equations

Consider the unsteady flow of  $N$  particles with mass  $M_i$ , ( $i = 1, \dots, N$ ) in a fluid with density  $\rho_f$  and viscosity  $\nu$ . Denote  $\Omega_f(t)$  as the domain occupied by the fluid at time  $t$ , and  $\Omega_i(t)$  as the domain occupied by the  $i$ th particle. So, the motion of an incompressible fluid is governed by the following Navier-Stokes equations in  $\Omega_f(t)$ ,

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

where  $\sigma$  is the total stress tensor in the fluid phase defined as

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

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.

The equations that govern the motion of each particle are the following Newton-Euler equations, i.e., the translational velocities  $\mathbf{U}_i$  and angular velocities  $\omega_i$  of the  $i$ th particle satisfy

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

where  $M_i$  is the mass of the  $i$ th particle ( $i = 1, \dots, N$ );  $\mathbf{I}_i$  is the moment of the inertia tensor;  $\Delta M_i$  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  $i$ th 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 section.  $\mathbf{F}_i$  and  $T_i$  are the resultants of the hydrodynamic forces

and the torque about the center of mass acting on the  $i$ th particle which are calculated by

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

where  $\boldsymbol{\sigma}$  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  $i$ th particle,  $\partial\Omega_i$  is the boundary of the  $i$ th 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  $i$ th particle and its angle  $\theta_i$  are obtained by integration of the kinematic equations

$$\frac{d\mathbf{X}_i}{dt} = \mathbf{U}_i, \quad \frac{d\theta_i}{dt} = \omega_i. \quad (5)$$

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

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

### 3 Collision Models

For handling more than one particle, a collision model is needed to prevent the particles from interpenetrating each other. Glowinski, Joseph and coauthors [8,9] proposed repulsive force models 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 these models, 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. [16] 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.

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 un-

avoidable numerical truncation errors. For the particle-particle collisions, the repulsive force is determined as,

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

where  $R_i$  and  $R_j$  are the radius of the  $i$ th and  $j$ th 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,  $\rho$  is the range of the repulsive force (usually  $\rho = 0.5 \sim 2.5\Delta h$ ,  $\Delta h$  is the mesh size),  $\epsilon_P$  and  $\epsilon'_P$  are small positive stiffness parameters for particle-particle collisions. If the fluid is sufficiently viscous, and  $\rho \simeq \Delta h$  as well as  $\rho_i/\rho_f$  are of order 1 ( $\rho_i$  is the density of the  $i$ th particle,  $\rho_f$  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,

$$\mathbf{F}_i^W = \begin{cases} 0, & \text{for } d'_i > 2R_i + \rho, \\ \frac{1}{\epsilon'_W}(\mathbf{X}_i - \mathbf{X}'_i)(2R_i - d'_i), & \text{for } d'_i \leq 2R_i, \\ \frac{1}{\epsilon_W}(\mathbf{X}_i - \mathbf{X}'_i)(2R_i + \rho - d'_i)^2, & \text{for } 2R_i \leq d'_i \leq 2R_i + \rho, \end{cases} \quad (8)$$

where  $\mathbf{X}'_i$  is the coordinate vector of the center of the nearest imaginary particle  $P'_i$  located on the boundary wall  $\Gamma$  w.r.t. the  $i$ th particle,  $d'_i = |\mathbf{X}_i - \mathbf{X}'_i|$  is the distance between the mass centers of the  $i$ th particle and the center of the imaginary particle  $P'_i$ .  $\epsilon_W$  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  $i$ th particle by the other particles and the walls can be expressed as follows,

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

#### 4 Multigrid FEM Fictitious Boundary Method

The details of multigrid FEM fictitious boundary method has been presented in Ref. [15,17,19]. For illustration, a brief description is given below.

The multigrid FEM fictitious boundary method (FBM) is based on a multigrid FEM background grid which covers the whole computational domain  $\Omega_T$  and can be chosen independently from the particles of arbitrary shape, size and number. It starts with a coarse mesh which may already contain many of the geometrical details of  $\Omega_i$ , ( $i = 1, \dots, N$ ), and it employs a fictitious boundary indicator (see [15]) 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 [17]). Hence, by making use of Eq. (6), we can perform calculations for the fluid in the whole domain  $\Omega_T$ . The considerable advantage of the multigrid FBM is that the total mixture domain  $\Omega_T$  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.,

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

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  $\alpha_i$ ,

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

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  $\alpha_i$  is zero everywhere except at the wall surface of the  $i$ th particle, and equals to the normal vector

$\mathbf{n}_i$  defined on the grid, i.e.,  $\mathbf{n}_i = \nabla\alpha_i$  (see also [14]). Then, the hydrodynamic forces acting on the  $i$ th particle can be computed by

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

The integral over each element covering the whole domain  $\Omega_T$  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  $i$ th particle, thus the volume integrals need to be computed only in one layer of mesh cells around the  $i$ th particle which leads to a very efficient treatment.

The algorithm of the multigrid FEM fictitious boundary method for solving 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 Eqs. (10) (a) and (b) in the corresponding fluid domain involving the position of the particles for the fictitious boundary conditions.
- (2) Calculate the corresponding hydrodynamic forces and the torque acting on the particles by using Eq. (12), 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. (10) (c) to set the new fluid domain and fictitious boundary conditions, and then advance to solve for the new velocity and pressure of the fluid phase as described in step (1).

## 5 Numerical Schemes

### 5.1 Time Discretization by Fractional-Step- $\theta$ Scheme

The fractional-step- $\theta$  scheme is a strongly A-stable time stepping approach, it possesses the full smoothing property which is important in the case of rough initial or boundary data. It also contains only very little numerical dissipation which is crucial in the computation of non-enforced temporal oscillations. A more detailed discussion of these aspects can be found in Ref. [21,18]. We first semi-discretize the Eqs. (10) (a) and (b) in time by the fractional-step- $\theta$  scheme. Given  $\mathbf{u}^n$  and the time step  $K = t_{n+1} - t_n$ , then solve for  $\mathbf{u} = \mathbf{u}^{n+1}$  and  $p = p^{n+1}$ . In the fractional-step- $\theta$ -scheme, one macro time step  $t_n \rightarrow t_{n+1} =$

$t_n + K$  is split into three consecutive substeps with  $\tilde{\theta} := \alpha\theta K = \beta\theta' K$ ,

$$\begin{aligned}
[I + \tilde{\theta}N(\mathbf{u}^{n+\theta})]\mathbf{u}^{n+\theta} + \theta K \nabla p^{n+\theta} &= [I - \beta\theta KN(\mathbf{u}^n)]\mathbf{u}^n \\
\nabla \cdot \mathbf{u}^{n+\theta} &= 0, \\
[I + \tilde{\theta}N(\mathbf{u}^{n+1-\theta})]\mathbf{u}^{n+1-\theta} + \theta' K \nabla p^{n+1-\theta} &= [I - \alpha\theta' KN(\mathbf{u}^{n+\theta})]\mathbf{u}^{n+\theta} \\
\nabla \cdot \mathbf{u}^{n+1-\theta} &= 0, \\
[I + \tilde{\theta}N(\mathbf{u}^{n+1})]\mathbf{u}^{n+1} + \theta K \nabla p^{n+1} &= [I - \beta\theta KN(\mathbf{u}^{n+1-\theta})]\mathbf{u}^{n+1-\theta} \\
\nabla \cdot \mathbf{u}^{n+1} &= 0,
\end{aligned} \tag{13}$$

where  $\theta = 1 - \frac{\sqrt{2}}{2}$ ,  $\theta' = 1 - 2\theta$ , and  $\alpha = \frac{1-2\theta}{1-\theta}$ ,  $\beta = 1 - \alpha$ ,  $N(\mathbf{v})\mathbf{u}$  is a compact form for the diffusive and convective part,

$$N(\mathbf{v})\mathbf{u} := -\nu \nabla \cdot (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \mathbf{v} \cdot \nabla \mathbf{u}. \tag{14}$$

Therefore, from Eq. (13) in each time step, we have to solve nonlinear problems of the following type,

$$[I + \theta_1 KN(\mathbf{u})]\mathbf{u} + \theta_2 K \nabla p = \mathbf{f}, \quad \mathbf{f} := [I - \theta_3 KN(\mathbf{u}^n)]\mathbf{u}^n, \quad \nabla \cdot \mathbf{u} = 0. \tag{15}$$

For the Eq. (10) (c), we simply take an explicit expression like,

$$\mathbf{u}^{n+1} = \mathbf{U}_i^n + \omega_i^n \times (\mathbf{X}^n - \mathbf{X}_i^n). \tag{16}$$

## 5.2 Space Discretization by Finite Element Method

If we define a pair  $\{\mathbf{u}, p\} \in H := \mathbf{H}_0^1(\Omega) \times L := L_0^2(\Omega)$ , and bilinear forms  $a(\mathbf{u}, \mathbf{v}) := (\nabla \mathbf{u}, \nabla \mathbf{v})$  and  $b(p, \mathbf{v}) := -(p, \nabla \cdot \mathbf{v})$ , a weak formulation of the Eq. (15) reads as follows,

$$\begin{cases} (\mathbf{u}, \mathbf{v}) + \theta_1 K [a(\mathbf{u}, \mathbf{v}) + n(\mathbf{u}, \mathbf{u}, \mathbf{v})] + \theta_2 K b(p, \mathbf{v}) = (\mathbf{f}, \mathbf{v}), & \forall \mathbf{v} \in H \\ b(q, \mathbf{u}) = 0, & \forall q \in L \end{cases} \tag{17}$$

here  $L_0^2(\Omega)$  and  $\mathbf{H}_0^1(\Omega)$  are the usual Lebesgue and Sobolev spaces,  $n(\mathbf{u}, \mathbf{u}, \mathbf{v})$  is a trilinear form defined by

$$n(\mathbf{u}, \mathbf{v}, \mathbf{w}) := \int_{\Omega} u_i \left( \frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} \right) w_j dx. \tag{18}$$



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 [21]:

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 [18]. 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 [22]).

In this paper, based on the latter approach combined with multigrid methods, we adopt the discrete projection method (DPM [22]) 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 [18]: 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 [18] for the details).

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 2). 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  $i$ th particle, its value is set to  $i$ . This array helps to reduce the CPU time for the volume integration of the force calculation.

Table 1  
Parameters for the meshes in the test calculation

Level	NVT	NMT	NEL	NEQ
3	222145	443328	221184	1107840
4	886657	1771392	884736	4427520
5	3542785	7081728	3538944	17702400

To evaluate these techniques regarding the CPU time when simulating particulate flows with large numbers of particles, we analyze the cases of 10 to

Table 2

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	3	4	5	3	4	5	3	4	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
<b>Total time</b>	<b>30</b>	<b>149</b>	<b>680</b>	<b>66</b>	<b>291</b>	<b>1400</b>	<b>486</b>	<b>1964</b>	<b>7954</b>
<b>Storage (MB)</b>	<b>4.8</b>	<b>19.5</b>	<b>78.0</b>	<b>4.8</b>	<b>19.5</b>	<b>78.0</b>	<b>4.8</b>	<b>19.5</b>	<b>78.0</b>

No. of Particles	= 10			= 1,000			= 100,000		
Level	3	4	5	3	4	5	3	4	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
<b>Total time</b>	<b>17</b>	<b>84</b>	<b>362</b>	<b>18</b>	<b>87</b>	<b>378</b>	<b>635</b>	<b>733</b>	<b>882</b>
<b>Storage (MB)</b>	<b>4.5</b>	<b>18.5</b>	<b>74.2</b>	<b>4.6</b>	<b>18.6</b>	<b>74.7</b>	<b>5.7</b>	<b>20.0</b>	<b>75.7</b>

100,000 particles falling down in a rectangular cavity filled with an incompressible Newtonian viscous fluid with and without these hierarchical techniques. In Table 1, "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 2 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.

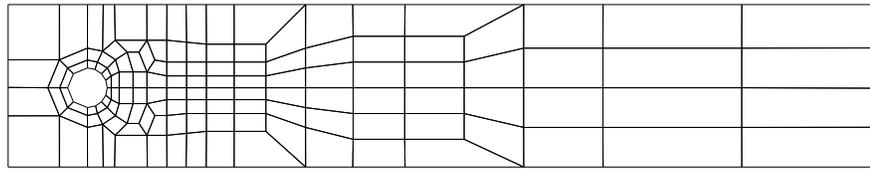
## 6 Numerical Experiments

In this section, we first test a series of simple configurations of benchmark character to evaluate and to validate the presented methodology. A careful comparison between the results obtained by the presented method and a standard body-fitted computation is 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. Then, one disk in a rotating circular container and one particle sedimenting in a fluid are examined to validate the calculated angular and translational velocities by the presented FBM. Finally, the simulations of three big particles plunging into 2000 small particles, and sedimentation of 10,000 particles in a cavity are given.

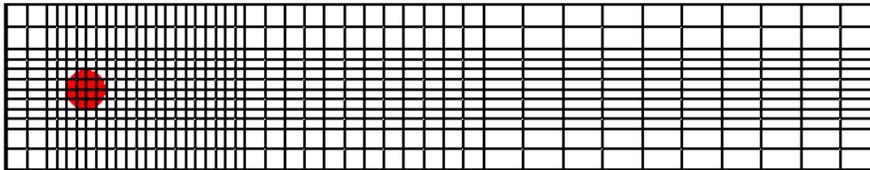
### 6.1 *Flow around a circular cylinder*

We first consider a benchmark case of flow around a fixed circular cylinder in a channel as described in [23]. Fig. 1 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  $H = 0.41$ , the cylinder diameter  $D = 0.1$ . 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  $\nu = \mu_f/\rho_f = 10^{-3}$  and its density by  $\rho_f = 1$ . The inflow profiles are parabolic  $U(0, Y, t) = 6.0\bar{U}Y(H-Y)/H^2$  with  $\bar{U} = 0.2$  such that the resulting Reynolds numbers are  $Re = 20$ . Table 3 and Table 4 give 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 [20]. 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  $\text{LEVEL} \geq 4$  on, we get an acceptable shape definition.



(a) Body-fitted mesh ( $\text{LEVEL} = 1$ )



(b) Cartesian mesh ( $\text{LEVEL} = 2$ ) for multigrid FBM

Fig. 1. Different coarse meshes adopted for flow around a fixed circular cylinder

Table 3

The parameters for sequentially refined body-fitted meshes

LEVEL	NVT	NMT	NEL	NEQ
1	156	286	130	702
2	572	1092	520	2704
3	2184	4264	2080	10608
4	8528	16848	8320	42016
5	33696	66976	33280	167232
6	133952	267072	133120	667264
7	534144	1066624	532480	2665728
8	2133248	4263168	2129920	10656256

Table 5 presents drag coefficient 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 that for the body-fitted mesh, especially when the ‘area ratio’ VEF is greater than 95%. The corresponding reference value of the drag coefficient  $C_d$  for this benchmark problem is also listed for comparison. It can be seen that there is a good agreement between the presented method and the reference computation. As expected, the results by the fixed rectangular mesh are a little bit worse than those obtained by the body-fitted mesh.

Table 4

The parameters for sequentially refined Cartesian meshes

LEVEL	NVT	NMT	NEL	NEQ	VEF(%)
1	161	292	132	716	63.662
2	585	1112	528	2752	95.493
3	2225	4336	2112	10784	95.493
4	8673	17120	8448	42688	97.482
5	34241	68032	33792	169856	99.472
6	136065	271232	135168	677632	99.721
7	542465	1083136	540672	2706944	99.814
8	2166273	4328960	2162688	10820608	99.953

Table 5

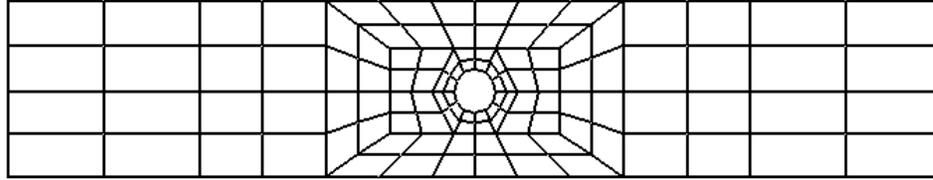
Drag coefficient  $C_d$  for flow around a circular cylinder with  $Re = 20$ 

LEVEL	Body-fitted mesh	Cartesian mesh
3	5.6645	5.3303
4	5.6001	5.4115
5	5.5844	5.4958
6	5.5808	5.5405
7	5.5799	5.5579
8	5.5799	5.5712
reference value	$C_d = 5.5795$	

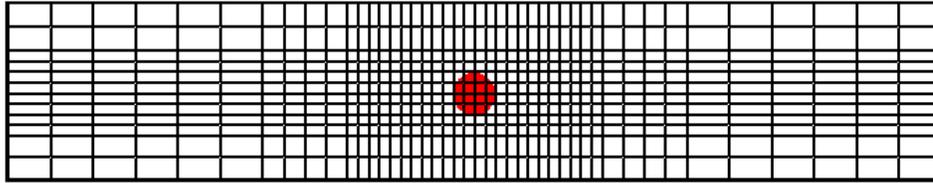
## 6.2 Moving cylinder with a prescribed velocity

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 [17])

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 channel grid. In Fig. 2, the body-fitted mesh is shown for the reference calculation while the fixed Cartesian mesh is taken for the presented FBM. Table 6 gives the parameters of the meshes in Fig. 2 with different numbers of refined levels.



(a) Body-fitted mesh (LEVEL = 2)



(b) Cartesian rectilinear grid (LEVEL = 2)

Fig. 2. Different ‘coarse’ meshes adopted for a moving cylinder in a channel

Table 6

Grid characteristics for sequentially refined meshes

	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

Fig. 3 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 coefficients  $C_d$  and  $C_l$  for one period after periodical flows are fully developed are shown in Fig. 3 (a) and (b), respectively, 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.

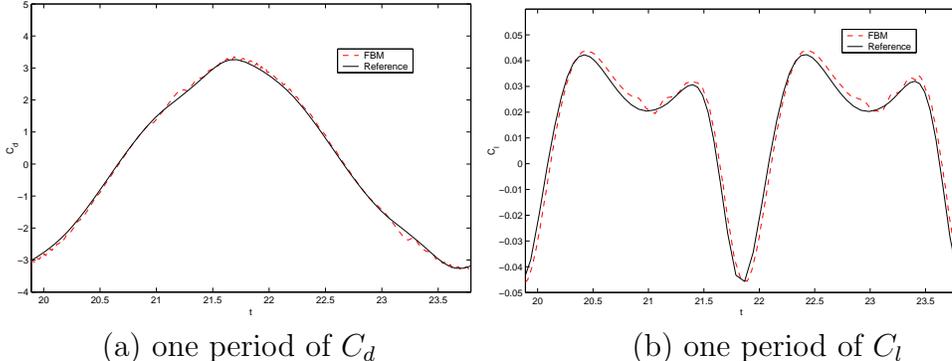
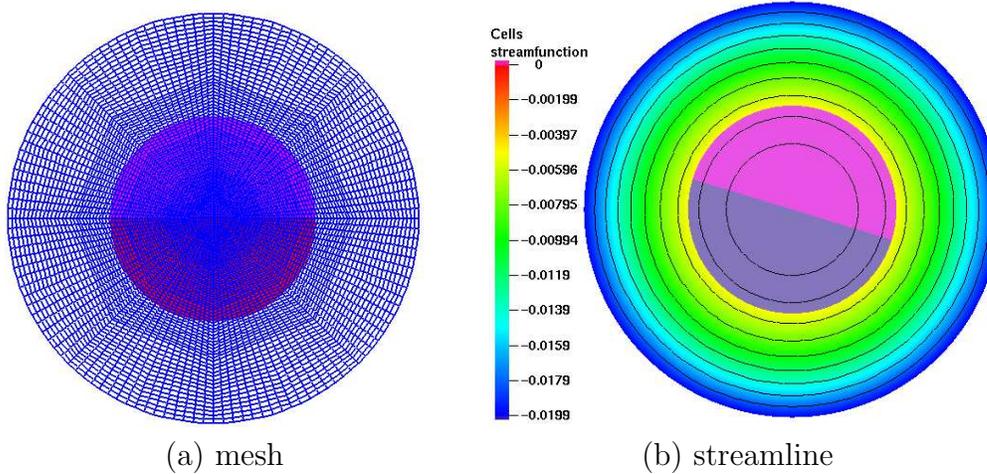


Fig. 3. Comparison of  $C_d$  and  $C_l$  between FBM and reference for a moving circular cylinder in a channel

### 6.3 One disk in a rotating circular container

We consider a circular disk with radius 1 centered in a circular container of radius 2 to validate the angular velocity, starting from rest. The boundary condition at the outside wall of the container imposes a rotation with an angular velocity  $\Omega = 0.01$ . Then, the disk should start rotating with the same angular speed, i.e., the steady solution is a rigid body rotation inside the container (including the disk) with the same angular velocity  $\Omega = 0.01$ . The mesh for the present calculation has 9,281 nodes and 9,216 elements. Fig. 4 (a) is the mesh adopted. Fig. 4 (b) shows 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 outside wall of the container. Moreover, we present the terminal angular velocity of the disk and the time needed to reach the steady limit with different viscosities. When the viscosity  $\nu$  becomes bigger, the disk can reach the steady state much faster. The presented results are identical with those for 3D case provided by Diaz-Goano, Minev and Nandakumar in Ref. [24].



viscosity $\nu$	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

Fig. 4. The motion of a circular disk in a rotating container and angular velocity at steady state

#### 6.4 One circular particle sedimenting in a channel

Next, we perform the numerical simulation of the motion of a circular particle sedimenting in an incompressible Newtonian viscous fluid to further validate the presented method. The computational domain is a channel of width 2 and height 6. A rigid circular particle 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  $\nu = 0.1$  or  $0.01$ . The gravity accelerating velocity  $g = 980$ . The diameter of the particle takes  $d = 0.25$ , and its density is chosen as  $\rho_p = 1.25$  or  $1.5$ . We suppose that the particle 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.

We carried out four case calculations corresponding to the fluid viscosity  $\nu = 0.1$  or  $0.01$  and the particle density  $\rho_p = 1.25$  or  $1.5$ . Table 7 gives the calculated maximum Reynolds number during the particle sedimenting in the fluid. The maximum Reynolds number is defined by  $Re = \sqrt{u^2 + v^2} \cdot d \cdot \rho_p / \nu$ ,

here  $u$  and  $v$  are the  $u$ -component and  $v$ -component velocity of the center of the particle, respectively. When the fluid viscosity decreases and the particle density increases, the maximum Reynolds number will increase. From the increased density of the particle and reduced viscosity of the fluid, we can see that the particle motion to be much faster and the symmetry breaking to be more pronounced. For cases of bigger fluid viscosity  $\nu = 0.1$ , the flows can be seen as laminar flows which have good results independent of mesh, while for cases of smaller fluid viscosity  $\nu = 0.01$ , the flows become unstable and turbulent, the results are much more dependent of mesh. It is not surprising since we are now dealing with a highly nonlinear phenomenon involving symmetry breaking. These results compare very well with those presented in Ref. [9].

Table 7

Maximum Reynolds number for a circular particle sedimenting in a fluid

Level	$\nu = 0.1$		$\nu = 0.01$	
	$\rho_p = 1.25$	$\rho_p = 1.5$	$\rho_p = 1.25$	$\rho_p = 1.5$
3	17.42	32.97	258.46	442.19
4	17.15	32.76	270.77	465.52

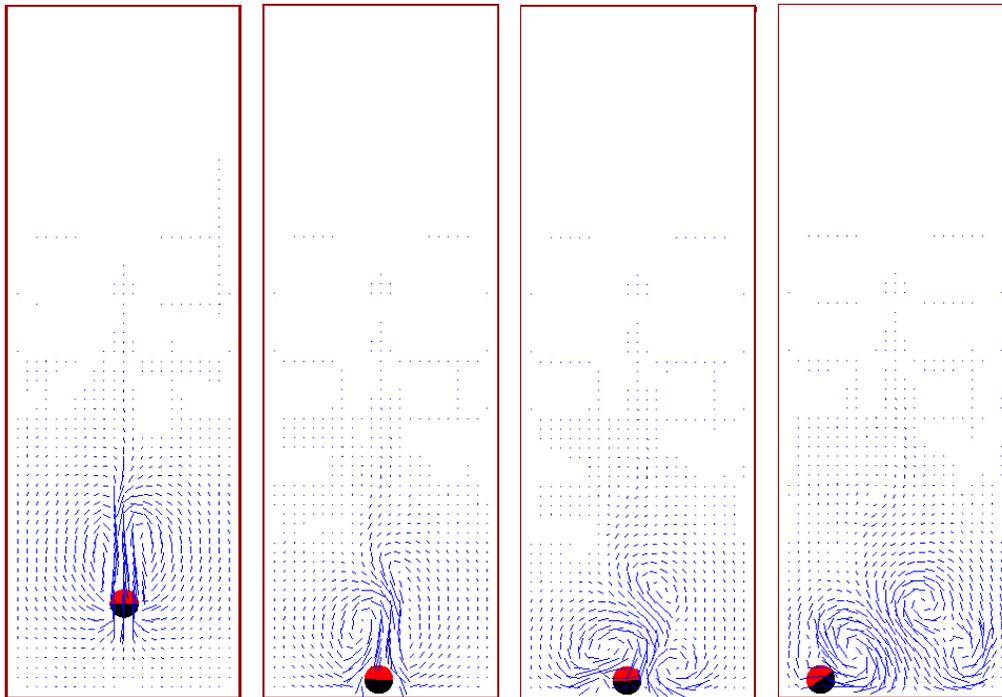


Fig. 5. One circular particle sedimenting in a fluid for the case of  $\nu = 0.01$  and  $\rho_p = 1.5$ : (a)  $t = 0.30$ , (b)  $t = 0.36$ , (c)  $t = 0.40$ , and (d)  $t = 0.50$  (from left to right).

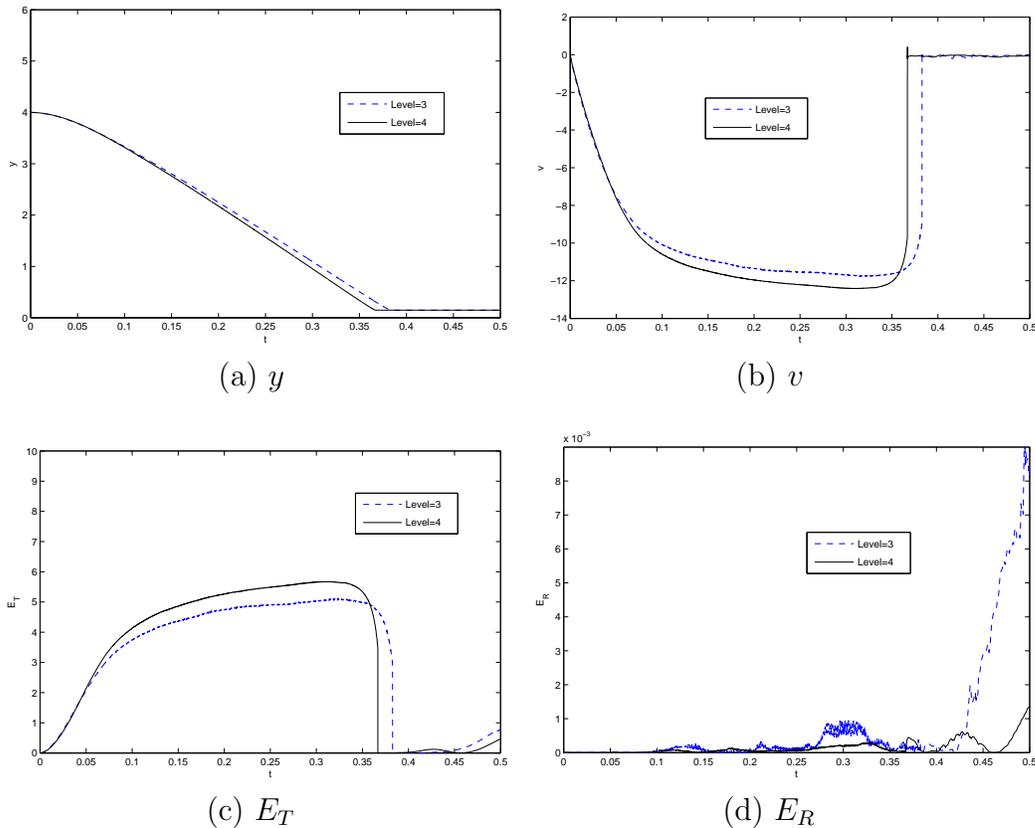


Fig. 6. One circular particle sedimenting in a fluid ( $\nu = 0.01$ ,  $\rho_p = 1.5$  and  $d = 0.25$ ): time histories of the  $y$ -coordinate (a) of the particle center,  $v$ -component (b) of the particle translational velocity of the particle center, translational kinetic energy(c), rotational kinetic energy (d), dashed line for  $\Delta h = 1/48$  (Level = 3), solid line for  $\Delta h = 1/96$  (Level = 4)

Fig. 5 shows snapshots of the velocity field for the case of  $\nu = 0.01$  and  $\rho_p = 1.5$ , computed with  $\Delta h = 1/96$  on Level = 4. We can see that at time  $t = 0.3$ , a symmetry breaking of small amplitude is taking place with the particle moving slightly on the left, away from the vertical symmetry axis of the cavity. After that, the particle quickly hits the bottom of the channel, the amplitude of symmetry breaking increases, and more complicated flows are formed under the perturbation of the falling particle. A careful examination of some quantities is presented in Fig. 6, including time histories of the  $y$ -coordinate of the particle center,  $v$ -component of the particle translational velocity of the particle center, translational kinetic energy ( $E_T = 0.5 M (u^2 + v^2)$ ,  $M$  is the mass of the particle), and the rotational kinetic energy ( $E_R = 0.5 \mathbf{I} \omega^2$ ,  $\mathbf{I}$  is the moment of the inertia of the particle,  $\omega$  is the angular velocity of the particle). 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 are easily disturbed by numerical errors.

### 6.5 Three big disks plunging into 2000 small particles

The following test problems differ significantly from the ones considered above 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.

The specific problem in this subsection is that three big circular disks plunge into 2000 small particles in a closed rectangular cavity. The position of the three big disks and the 2000 small particles at time  $t = 0$  is shown in Fig. 7 (a).

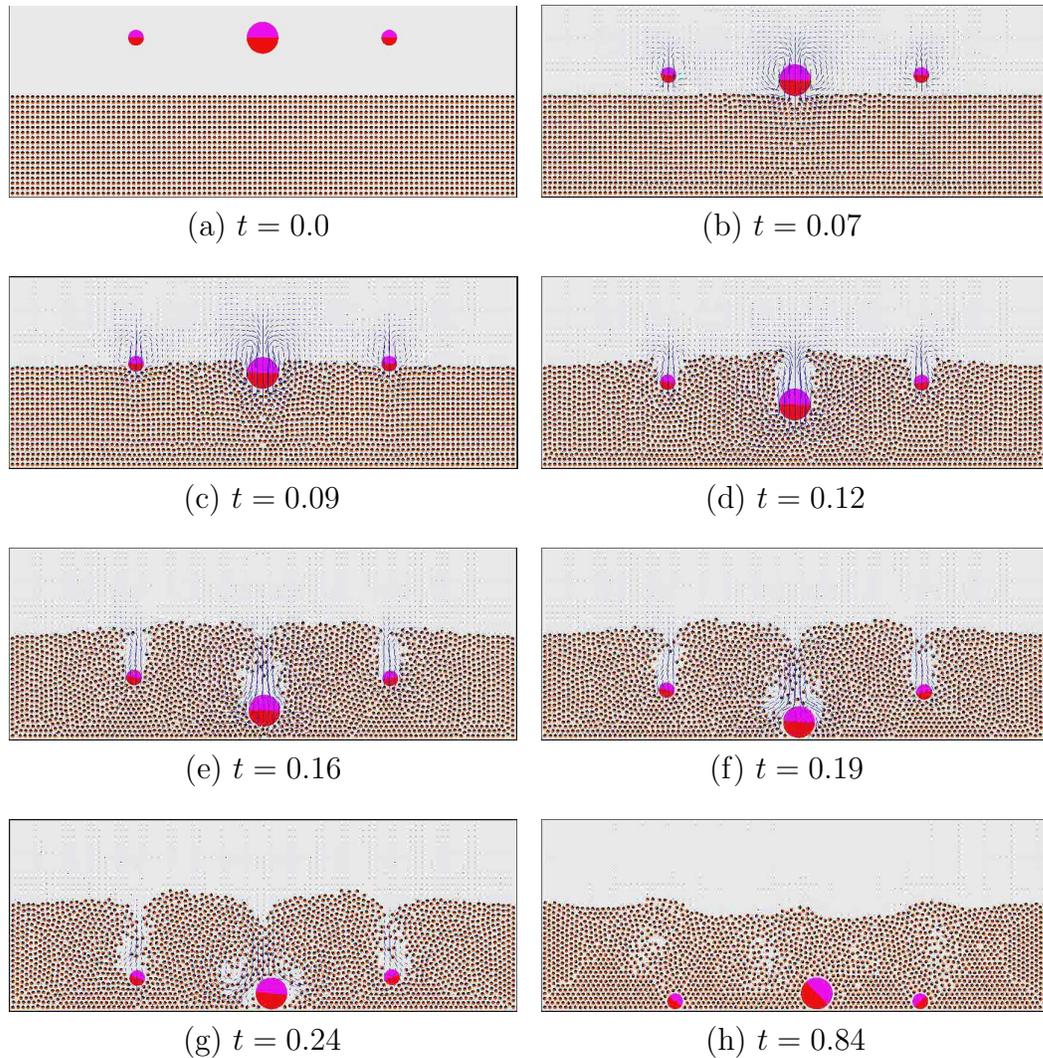


Fig. 7. Snapshots of velocity field for three big disks plunging into 2000 small particles in a 2D cavity

The width and height of the cavity are 8 and 3. The three big disks are located

at upper of the cavity with their center positions of (2.0, 2.5), (4.0, 2.5) and (6.0, 2.5), as well as their diameters of 0.25, 0.5 and 0.25, respectively. The density of these three big disks is 2.0. The 2000 small particles are placed at the bottom of the cavity with 20 rows, and in each row there are 100 particles with diameter of 0.0595 and density of 1.1, respectively. The disks, particles and fluid are at rest at  $t = 0$ . An uniform mesh with mesh size  $\Delta h = 0.0208$  of 55825 nodes and 55296 elements (LEVEL = 3) is adopted. The range of the repulsive force is chosen as  $\rho = 0.02$ . The density of the fluid is  $\rho_f = 1$ . The viscosity of the fluid is  $\nu = 10^{-3}$ . The gravity accelerating velocity  $g = 980$ . The parameter  $\epsilon_P$  in the collision model has been taken equal to  $5 \times 10^{-7}$ , and  $\epsilon_W = \epsilon_P/2$ ,  $\epsilon'_P = \epsilon_P$ ,  $\epsilon'_W = \epsilon_W$ .

The snapshots of the evolution and velocity field for three big disks plunging into 2000 small particles are shown in Fig. 7. At time  $t = 0.07$ , the biggest disk at the middle first touches the below small particles; at time  $t = 0.09$ , the other two disks also touch the small particles, while the biggest disk pushes off the small particles and digs into them. At time  $t = 0.012$ , all three disks gouge into the small particles and at the same time, there are three hollows surrounding with the small particles above the corresponding three disks being formed; after that, the three big disks continue to fall down, push the small particles away, until they hit the bottom of the cavity, accompanying by the closure of the three hollows and filling the hollows inside with the small particles completely. It can be seen that some irregular waves generated on the interface between the small particles and fluid under the disturbance of the three big disks. Actually, this case (solid-liquid two phase flow) is very similar as that for free surface flows with two liquid phases. We believe that the presented simulating results and methods can help to understand more other multiphase flows.

### 6.6 Sedimentation of 10,000 circular particles

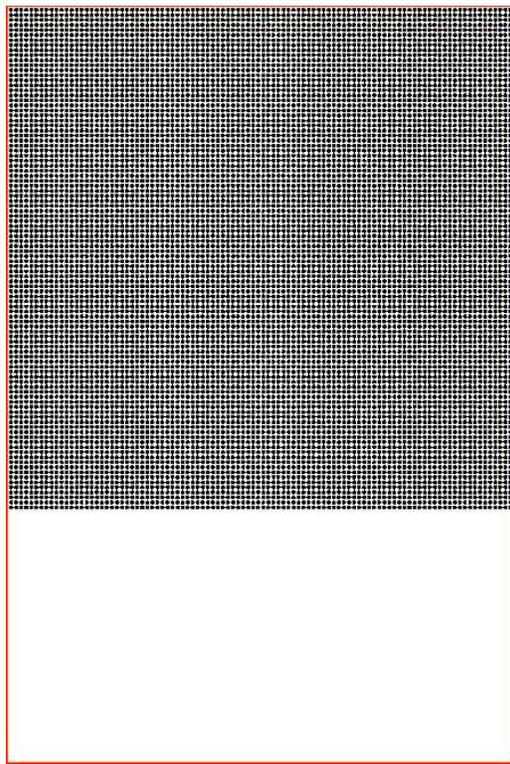
Finally, we consider the sedimentation of 10,000 circular particles with identical size falling down in a closed rectangular cavity. The width and height of the cavity are 8 and 12. In the case, there are 100 rows and in each row there are 100 particles. The 10,000 particles are placed at the top of the cavity. The diameter of the particles is 0.0693. We have chosen an uniform mesh with mesh size  $\Delta h = 0.0208$  of 222145 nodes and 221184 elements (LEVEL = 3). The range of the repulsive force is chosen as  $\rho = 0.01$ . The position of the particles at time  $t = 0$  is shown in Fig. 8 (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  $\rho_i = 1.1$  ( $i = 1, \dots, 10000$ ). The viscosity of the fluid is  $\nu = 10^{-2}$ . The gravity accelerating velocity  $g = 980$  (all quantities in non-dimensional form). The parameter  $\epsilon_P$  in the collision model has been taken equal to  $5 \times 10^{-5}$ , and

$$\epsilon_W = \epsilon_P/2, \epsilon'_P = \epsilon_P, \epsilon'_W = \epsilon_W.$$

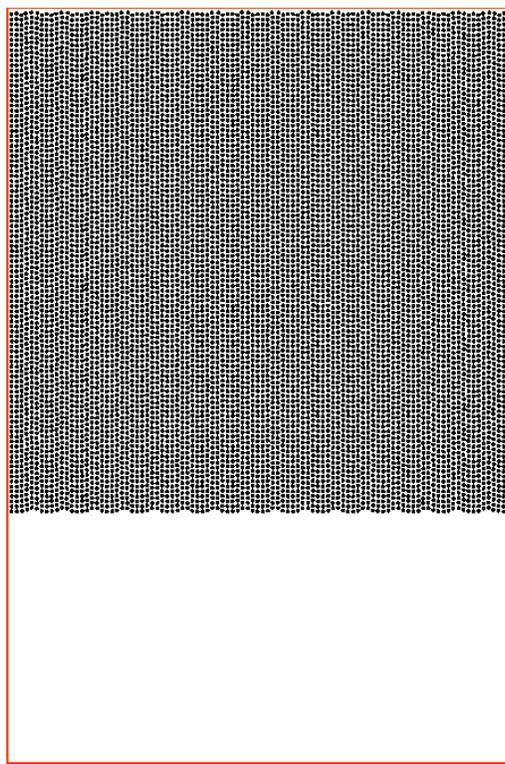
The snapshots of the evolution and velocity fields for the sedimentation of 10,000 circular particles are shown in Figs. 8, 9 and 10. We can see that the simulation gives rise to fingering which resembles Rayleigh-Taylor instability (see Figs. 8 (b), (c) and (d)). The waves have a well defined wavelength and growth rate which we shall model as a conventional Rayleigh-Taylor instability of heavy fluid above light. Fig. 9 clearly shows the development of the Rayleigh-Taylor instability, 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". In Fig. 10, we can see that 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.

## 7 Conclusions

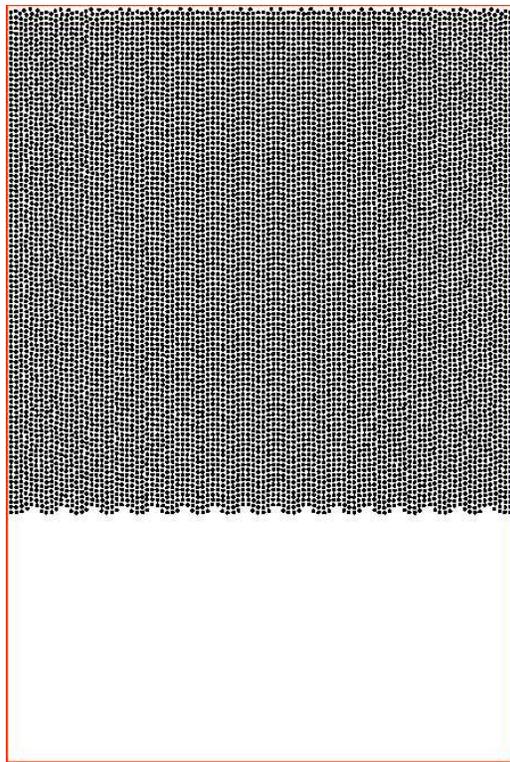
We have presented the multigrid FEM fictitious boundary method (FBM) for the direct numerical simulation of solid-liquid two phase flows with large number of moving particles in 2D. The presented method treats the fluid part, the calculation of forces and the movement of the particles in a subsequent manner, which is computationally cheap and simple to implement. Its accuracy has been proven by a series of comparisons between the presented results and corresponding reference results from own computations or from the literature. The result that it possesses good potentialities to efficiently simulate real particulate flows with huge number of particles is shown by two numerical examples of three big particles plunging into 2000 small particles and sedimentation of 10,000 particles in a cavity, although complex configurations with numerous particles typically require small time steps by physical and numerical stable reasons, and also the comparison of total efficiency with more implicit scheme, for instance [9,10], remains further investigation. Moreover, the presented method 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. Finally, the presented 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.



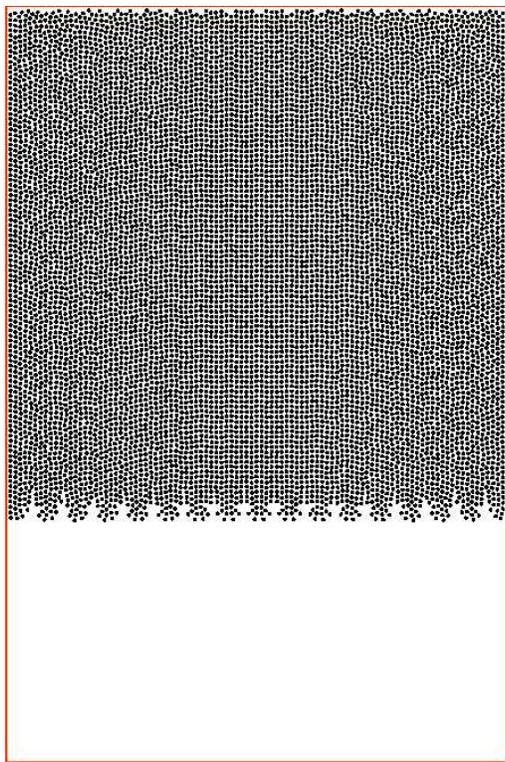
(a)  $t = 0.0$



(b)  $t = 0.15$

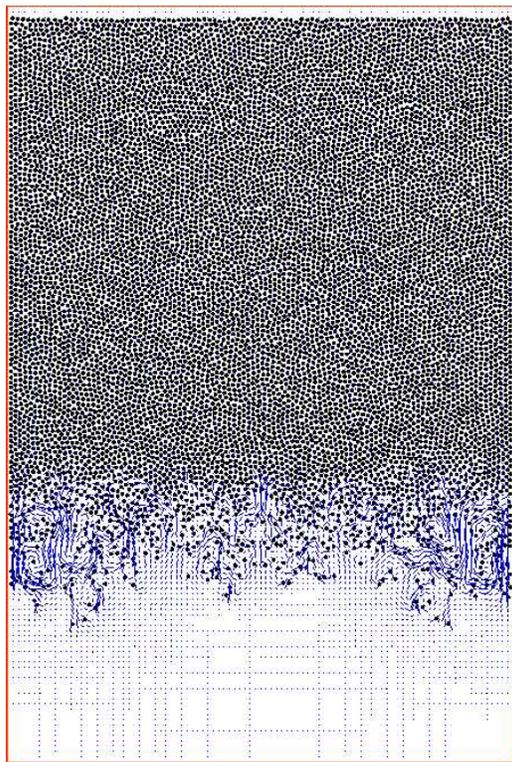


(c)  $t = 0.21$

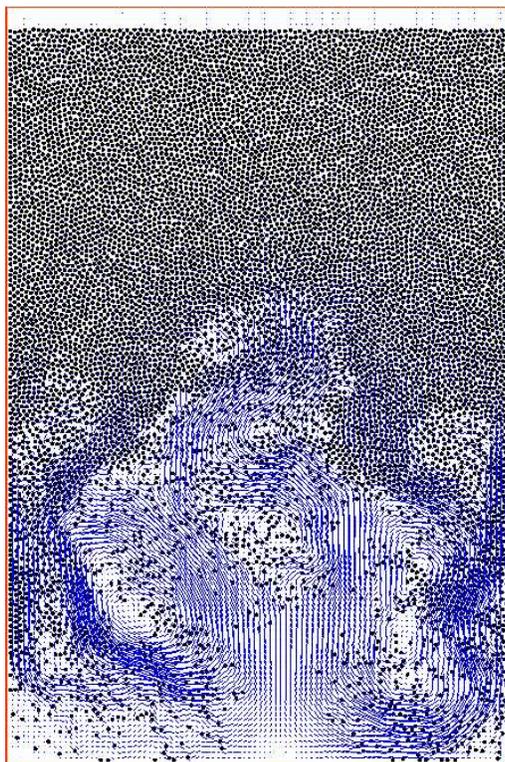


(d)  $t = 0.30$

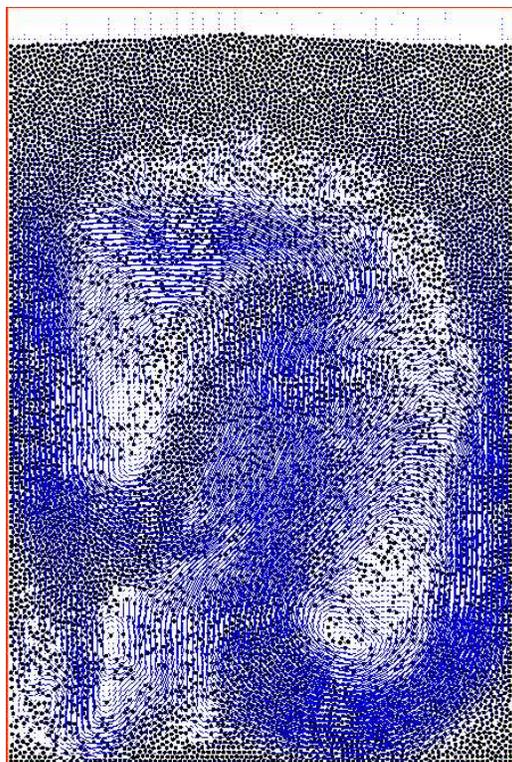
Fig. 8. Snapshots of fingering phenomena during the sedimentation of 10,000 particles in a 2D cavity



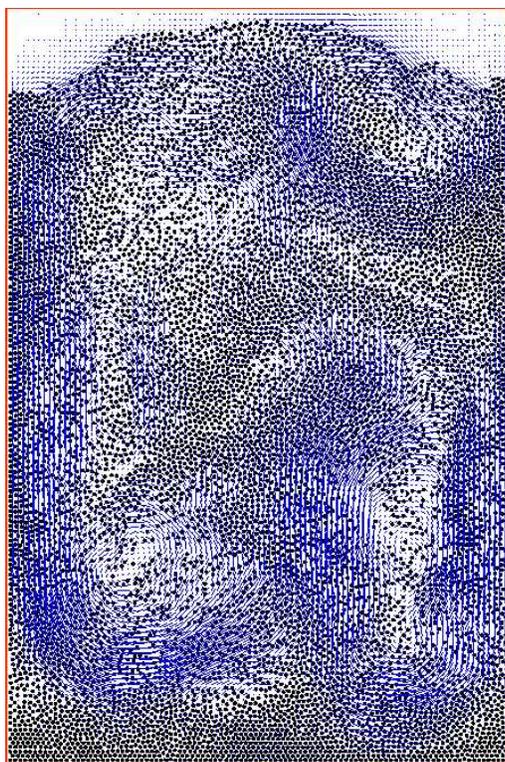
(a)  $t = 1.0$



(b)  $t = 2.0$

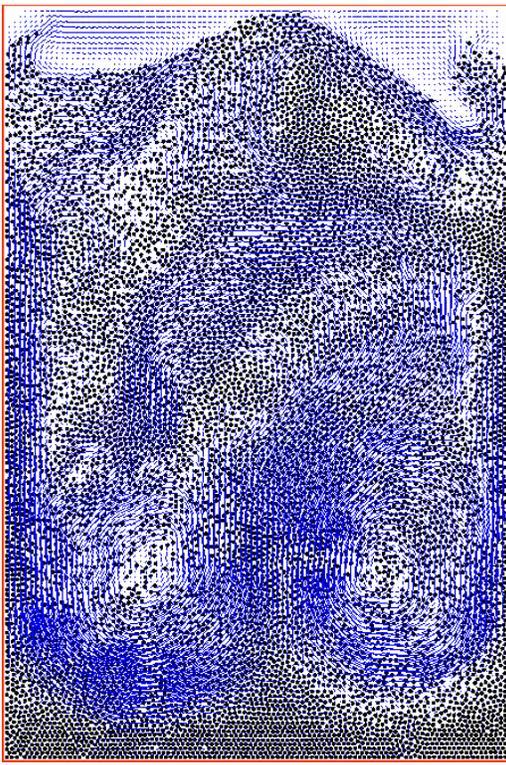


(c)  $t = 3.0$

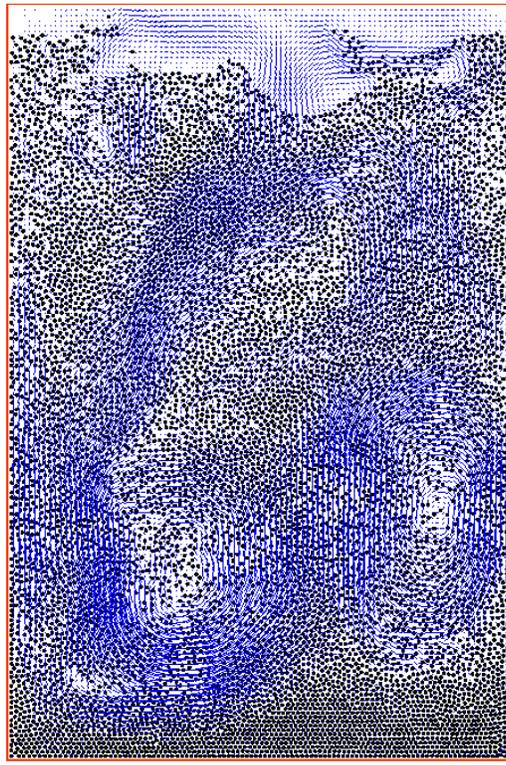


(d)  $t = 4.0$

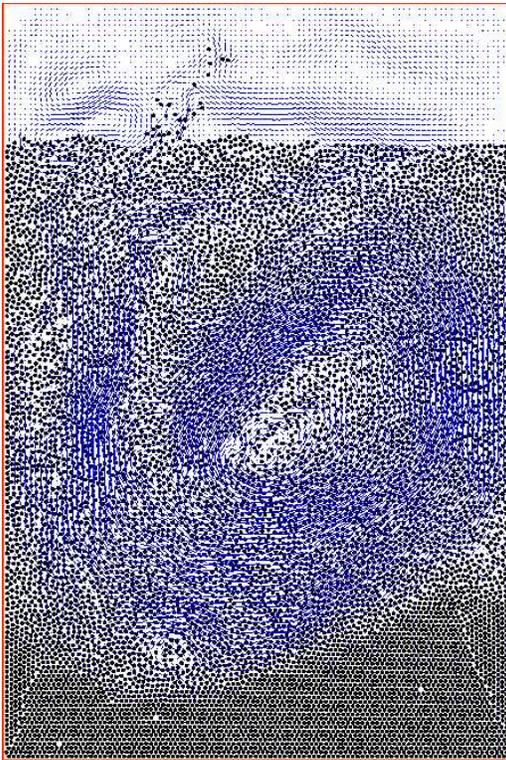
Fig. 9. Snapshots of velocity field for the sedimentation of 10,000 particles in a 2D cavity (continued)



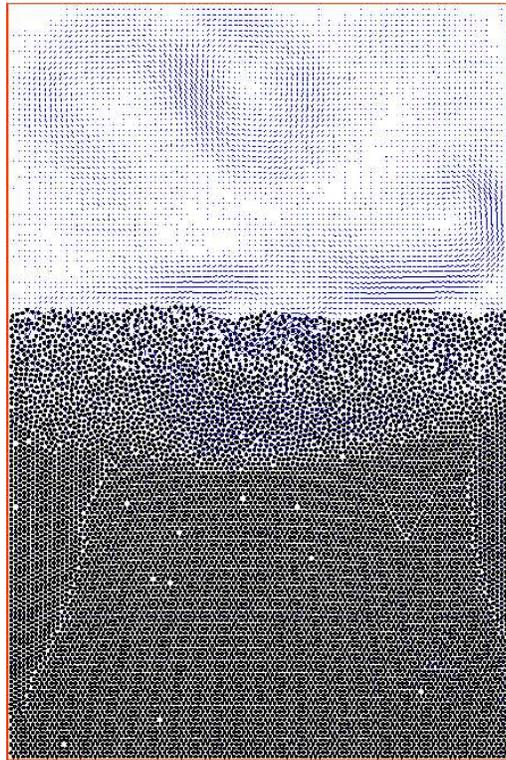
(a)  $t = 4.5$



(b)  $t = 5.0$



(c)  $t = 8.0$



(d)  $t = 16.5$

Fig. 10. Snapshots of velocity field for the sedimentation of 10,000 particles in a 2D cavity (continued)

## References

- [1] Hu, H.H., Joseph, D.D. and Crochet, M.J.: Direct simulation of fluid particle motions, *Theor. Comp. Fluid Dyn.* **3**, (1992), 285-306
- [2] Hu, H.H.: Direct simulation of flows of solid-liquid mixtures, *Int. J. of Multiphase Flow.* **22**, (1996), 335-352
- [3] Hu, H.H., Patankar, N.A. and Zhu, M.Y.: Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian-Eulerian techniques, *J. Comput. Phys.*, **169**, (2001), 427
- [4] 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
- [5] Maury, B.: Characteristics ALE method for the unsteady 3D Navier-Stokes equations with a free surface, *Int. J. of Comput. Fluid Dynamics*, **6**, (1996), 175-188
- [6] Maury, B.: Direct Simulations of 2D fluid-particle flows in biperiodic domains, *J. Comput. Phys.*, **156**, (1999), 325-351
- [7] 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
- [8] 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. Phys.*, **169**, (2001), 363-426
- [9] 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)
- [10] 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
- [11] Ladd, A. J. C.: Numerical simulation of particulate suspensions via a discretized Boltzmann equation. I. Theoretical foundation. *J. Fluid Mech.*, **271**, (1994), 285-309
- [12] Ladd, A. J. C.: Numerical simulation of particulate suspensions via a discretized Boltzmann equation. II. Numerical results. *J. Fluid Mech.*, **271**, (1994), 311-339
- [13] Ladd, A. J. C. and Verberg, R.: Lattice-Boltzmann simulations of particle fluid suspensions. *J. Stat. Phys.*, **104**, (2001), 1191
- [14] Duchanoy, C. and Jongen, T.R.G.: Efficient simulation of liquid-solid flows with high solids fraction in complex geometries. *Computers and Fluids* **32**, (2003), 1453

- [15] 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
- [16] 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
- [17] 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
- [18] Turek, S.: *Efficient Solvers for Incompressible Flow Problems*. Springer Verlag, Berlin-Heidelberg-New York, (1999)
- [19] Wan, D.C. and Turek, S.: Direct Numerical Simulation of Particulate Flow via Multigrid FEM Techniques and the Fictitious Boundary Method, *Int. J. Numer. Method in Fluid*, (2004), under review.
- [20] Turek, S.: **FEATFLOW** *Finite element software for the incompressible Navier–Stokes equations: User Manual, Release 1.1*, University of Heidelberg, (1998)
- [21] 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
- [22] Turek, S.: On discrete projection methods for the incompressible Navier–Stokes equations: An algorithmical approach, *Comput. Methods Appl. Mech. Engrg.*, 143, (1997), 271 - 288
- [23] 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
- [24] Diaz-Goano, C., Mineev, P.D. and Nandakumar, K.: A fictitious domain/finite element method for particulate flows, *J. Comput. Phys.*, **192**, (2003), 105-123