A new Immersed Boundary and Moving Mesh Method for Particulate Flow

Dan Anca and Stefan Turek

Dortmund University, Mathematics, LS III, Institute of Applied mathematics
Vogelpotsweg 87, Dortmund, 44225, Germany
E-mail adan@mathematics.uni-dortmund.de
ture@featflow.de

Keywords: Particulate Flow, FEM, Fictitious Boundary Method, Moving Mesh Techniques, Multigrid Solvers

Abstract

This paper discusses numerical simulation techniques for particulate flow using an efficient immersed boundary method coupled with a new moving mesh approach. The immersed boundary method is represented by an explicit multigrid fictitious boundary method (FBM) based on an (unstructured) FEM background grid. We compute the flow using an ALE formulation, while the solid particles of arbitrary size and shape can move freely through the fluid. The computational mesh can be additionally aligned by a special moving mesh method to achieve a higher accuracy near the interfaces. Numerical simulations demonstrate the high accuracy and efficiency of the presented method.

Introduction

Numerical simulation of rigid particulate flows or the motion of small rigid particles in a viscous liquid is one of the main focuses of engineering research and still a challenging task in many applications. Depending on the area of application, these types of problems arise frequently in numerous settings, such as sedimentation and fluidized suspensions, lubricated transport, hydraulic fracturing of reservoirs, slurries, understanding solid-liquid interaction, etc.

Several numerical simulation techniques for the particulate flows have been developed over the past decade. In these methods, the fluid flow is governed by the continuity and momentum equations, while the particles are governed by the equation of motion for a rigid body. The flow field around each individual particle has to be resolved, such that the hydrodynamical force between the particle and the fluid can be obtained from the solutions. Hu, Joseph and coworkers. (1; 2), Galdi (3) and Maury (4) developed finite element methods based on unstructured grids to simulate the motion of large number of rigid particles in Newtonian and viscoelastic fluids. This approach is based on an arbitrary Lagrangian-Eulerian (ALE) technique. Both the fluid and solid equations of motion are incorporated into a single coupled variational equation. The method developed by them does not take in account torques and hydrodynamical forces. At each time step, a new mesh is generated when the old one becomes too distorted, and the flow field is projected on to the new mesh. 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. However this method showed a limited possibility in the 3D case.

Moreover, Glowinski, Joseph, Patankar and coauthors (5; 6; 7; 8) proposed a distributed Lagrange multiplier (DLM)/fictitious domain method for the direct numerical simulation of large number of rigid particles in fluids. In the 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. This formulation permits the use of a fixed structured grid thus eliminating the need for remeshing the domain. Our group (9; 10; 11; 12) presented another multigrid fictitious boundary method (FBM) for the detailed simulation of particulate flows. The method is based on a fixed unstructured FEM background grid. The motion of the solid particles is modelled by the Newton-Euler equations. Based on the boundary conditions applied at the interface between the particles and the fluid which can be seen as an additional constraint to the governing Navier-Stokes equations, the fluid domain can be extended into the whole domain which covers both fluid and particle domains. The FBM starts with a coarse mesh which may contain already many of the geometrical fine-scale details, and employs a (rough) boundary parametrization which sufficiently describes all large-scale structures with regard to the boundary conditions. Then, all fine-scale features are treated as interior objects such that the corresponding components in all matrices and vectors are unknown degrees of freedom which are implicitly incorporated into all iterative solution steps. An adavantage of these fictitious domain methods over the generalized standard Galerkin finite element method is that the fictitious domain methods allow a fixed grid to be used, eliminating the need for remeshing and they can be handled independently from the flow features. Much progress has been made for adopting the fictitious domain methods to simulate the particulate flows, yet the quest for more accurate and efficient methods remains active. As we know, an underlying problem when adopting the fictitious domain methods is that the boundary approximation is of low accuracy. Particulately in three space dimensions, the ability of the fictitious domain methods to deal with the interaction between fluid and rigid particles accurately is greatly limited. One remedy could be to preserve the mesh topology, for instance as generalized tensorproduct or blockstructured meshes, while a local alignment with the physical boundary of the solid is achieved by a moving mesh process, such that the boundary approximation error can be significantly decreased.

For obtaining improvements in accuracy and efficiency, the adaptive mesh method prooved to be a very powerfool tool. There are many existing mesh adaptive methods to achieve this purpose. Generally speaking, mesh adaptivity is usually in the form of local mesh refinements or through a continuous mesh mapping. In local adaptive mesh refinement methods (13), an adaptive mesh is obtained by adding or removing points to achieve a desired level of accuracy. This allows a systematic error analysis. However, local refinement methods require complicated data structures and fairly technical methods to communicate information among different levels of refinements. In the mapping approach (14; 15), the mesh points are moved continuously in the whole domain to concentrate in regions where the solution has the largest variations or moving interfaces locate. These solution-adaptive or geometry-adaptive mesh maps can be used to compute accurately the sharp variation or the moving interface problems. They also have the additional advantage of allowing the use of standard solvers as all the computations are performed in the logical domain using a uniform mesh. Over the past decade, several mesh adaptive techniques have been developed, namely the so-called h-, p- and r- methods. The first two do static regridding with fixed time, where the h-method does automatic refinement or coarsening of the spatial mesh based on a posteriori error estimates or error indicators and the p-method takes higher or lower order approximations locally as needed. In contrast, the r-method (also known as moving mesh method) relocates grid points in a mesh having a fixed number of nodes in such a way that the nodes remain concentrated in regions of rapid variation of the solution or corresponding moving interfaces. The r-method is a dynamic method which means that it uses time stepping or pseudo-time stepping approaches to construct the desired transformation. The r-method or moving mesh method differs from the h-, and p-methods in that the former keeps the same number of mesh points throughout the entire solution process, while the later have to treat the tedious hanging node problems. Thus, the size of computation and data structure are fixed, which enables the r-method much easier to incorporate into most CFD codes without the need for changing of system matrix structures and special interpolation procedures. The r-method has received more and more attention due to some new developments which clearly demonstrate its potential for problems such as those having moving interfaces (16; 17; 18; 19; 20).

The primary objective of this paper is to combine the multigrid fictitious boundary method (FBM) (9; 11) with the moving mesh method described in (20) for the simulation of the particulate flows to check the accuracy of the proposed combining method comparing its results with the pure multigrid fictitious boundary method (FBM). As we have shown in (11), the use of the multigrid FBM does not require to change the mesh during the simulations, although the rigid 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 surfaces of solid particles is only of first order which might lead to accuracy problems. For a better approximation of the particle surfaces, we adopt a deformed grid, created from an equidistant cartesian mesh, in which the topology is preserved, only the grid spacing is changed such that the grid points are concentrated near the surfaces of the rigid particles. Only the solution of additional linear Poisson problems in every time step is required for generating the deformation grid, which means that the additional work is significantly less than the main fluid-particle part. The paper is organized as follows. In Section 3, the physical models together with collision and agglomeration models for the rigid particulate flows are described. The detailed numerical schemes including the multigrid FBM and the moving mesh method are given in Section 4. Numerical experiments are implemented and their computational results will be presented in Section 5. The concluding remarks will be given in Section 6.

Nomenclature

We use in our approach non-dimensional formulas. However, the notations of the variables are choosen in such a way that a connection with physics could be made.
In the section of **Governing Equations** the following notations are used:

u	velocity of the fluid
t	time
p	pressure
n	normal vector
p	pressure
M_i	mass of the particle
U_i	translational velocity of a particle
g	gravitational acceleration
$\mathbf{I}_i$	moment of inertia of a particle
F_i	hydrodinamical force acting on a particle
T_i	torque acting on a particle
X	position of the particle

Greek letters

Ω_f	domain occupied by the fluid
Ω_i	domain occupied by a particle
Ω_T	total domain ($\Omega_T = \Omega_f \bigcup \Omega_i$)
ρ_f	density of the fluid
ν	viscosity
σ	total stress tensor
ω_i	angular velocity of a particle
θ_i	angle of a particle

In the section of **Collision and Agglomeration Models** the following notations are used:

$F_{i,j}^P$	repulsive force between two particles
F_i^W	repulsive force between one particle and the wall
R	radius of the particles
$d_{i,j}$	distance between two particles
$d_i^{'}$	distance between one particle and the wall
$F_i^{'}$	total repulsive force acting on the i-th particle

Greek letters

ρ	repulsive range
ε_P	stiffness parameter for particle-particle collision
ε_W	stiffness parameter for particle-wall collision

In the section of **Moving Mesh Method**, new notations come in:

$f(x)$	monitor function
$g(x)$	area function
c_f and c_g	scale factor for monitor function
$v(x)$	grid-velocity vector

Other important notations are:

α	Dirac function
W_m	mesh velocity

Description of the Physical Models

Governing Equations

In our numerical studies of particle motion in a fluid, we will assume that the fluids are immiscible and Newtonian. The particles are assumed to be rigid. Let us consider the unsteady flow of N particles with 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 motion of an incompressible fluid is governed by the Navier-Stokes equations in $\Omega_f(t)$,

$$\begin{aligned} &\rho_f \left(\tfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) - \nabla \cdot \sigma = 0 \\ &\nabla \cdot \mathbf{u} = 0 \end{aligned} \tag{1}$$

for $t \in (0, T)$ and 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 time. 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 ω_i of the ith particle satisfy

$$\begin{aligned} &M_i \tfrac{d\,\mathbf{U}_i}{d\,t} = (\Delta M_i)\,\mathbf{g} + \mathbf{F}_i + \mathbf{F}'_i \\ &\mathbf{I}_i \tfrac{d\,\omega_i}{d\,t} + \omega_i \times (\mathbf{I}_i\,\omega_i) = T_i \end{aligned} \tag{3}$$

where M_i is the mass of the ith particle; $\mathbf{I}_i$ 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

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

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

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

Collision and Agglomeration Models

Following different models, we examine a special collision model with a new definition of short range repulsive forces which can prevent the particles from getting too close and can also deal with the case of particles overlapping 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,

$$\mathbf{F}_{i,j}^P = \begin{cases} 0, & (a) \\ \frac{1}{\epsilon'_P}(\mathbf{X}_i - \mathbf{X}_j), & (b) \\ \frac{1}{\epsilon_P}(\mathbf{X}_i - \mathbf{X}_j), & (c) \end{cases} \tag{7}$$

where

$$\begin{aligned}&(a) \Leftrightarrow d_{i,j} > R_i + R_j + \rho \\ &(b) \Leftrightarrow d_{i,j} < R_i + R_j \\ &(c) \Leftrightarrow R_i + R_j \leq d_{i,j} \leq R_i + R_j + \rho.\end{aligned} \qquad (8)$$

Here, R_j are the radius of the ith and jth particle, $\mathbf{X}_i$ and $\mathbf{X}_j$ are the coordinates of their mass centers, $d_{i,j} = |\mathbf{X}_i - \mathbf{X}_j|$ is the distance between their 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,

$$\mathbf{F}_i^W = \begin{cases} 0, & (a\prime) \\ \frac{1}{\epsilon'_W}(\mathbf{X}_i - \mathbf{X}'_i)(2R_i - d'_i), & (b\prime) \\ \frac{1}{\epsilon_W}(\mathbf{X}_i - \mathbf{X}'_i)(2R_i + \rho - d'_i)^2, & (c\prime) \end{cases} \qquad (9)$$

where

$$\begin{aligned}&(a\prime) \Leftrightarrow d'_i > 2R_i + \rho \\ &(b\prime) \Leftrightarrow d'_i < 2R_i \\ &(c\prime) \Leftrightarrow 2R_i \leq d'_i \leq 2R_i + \rho\end{aligned} \qquad (10)$$

$\mathbf{X}'_i$ 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,

$$\mathbf{F}'_i = \sum_{j=1, j\neq i}^{N} \mathbf{F}^P_{i,j} + \mathbf{F}^W_i. \qquad (11)$$

Future plans for this research include also a development of an agglomeration model. As a first step on this direction, we performed a trick to the actual collison model described above such that we obtained an agglomeration model. This trick is nothing else then switching the sign for the forces defined on the collision model. So, instead of a positive sign we put a negative one and in this manner the repulsive forces become attractive forces. The result is that the particles will not separate anymore when they touch each other, but will stick together during the moment of touching. Lately in this paper we will provide results obtained in this sense of agglomeration. However, an clear and stable agglomeration model should be defined further on.

Numerical Method

Multigrid FEM Fictitious Boundary Method

The details of multigrid FEM fictitious boundary method have been presented in Refs. (9; 10; 11). 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 Ω_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 Ω_i $(i = 1, \ldots, N)$, and it employs a fictitious boundary indicator (see (9)) 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 (10)). 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.,

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 & (a) \\ \rho_f \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \sigma = 0 & (b) \\ \mathbf{u}(\mathbf{X}) = \mathbf{U}_i + \omega_i \times (\mathbf{X} - \mathbf{X}_i) & (c) \end{cases} \qquad (12)$$

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,

$$\alpha_i(\mathbf{X}) = \begin{cases} 1 & \text{for} \quad \mathbf{X} \in \Omega_i \\ 0 & \text{for} \quad \mathbf{X} \in \Omega_T \setminus \Omega_i \end{cases} \qquad (13)$$

where $\mathbf{X}$ denotes the coordinates. The importance of such a definition 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 $\mathbf{n}_i$ of wall surface of the ith particle defined on the grid, i.e., $\mathbf{n}_i = \nabla \alpha_i$. Then, the hydrodynamic forces acting on the ith particle can be computed by

$$
\begin{aligned}
&\mathbf{F}_i = -\int_{\Omega_T} \sigma \cdot \nabla\alpha_i \, d\Omega \\
&T_i = -\int_{\Omega_T} (\mathbf{X} - \mathbf{X}_i) \times (\sigma \cdot \nabla\alpha_i) \, d\Omega.
\end{aligned} \tag{14}
$$

The integral over each element covering the whole domain Ω_T can be exactly calculated with a standard Gaussian quadrature of sufficiently high order (4×4 points). 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.

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. (12) (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. (14), and compute the collision forces by Eq. (11).

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. (12) (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.

Moving Mesh Method

In this subsection, we briefly describe the moving mesh method which will be adopted and coupled with above multigrid fictitious boundary method (FBM) to solve numerically the particulate flows. The details of the moving mesh method can be found in Ref. (20).

The moving mesh problem can be equated to construct a transformation φ, $x = \varphi(\xi)$ from computational space (with coordinate ξ) to physical space (with coordinate x). There are two basic types of moving mesh methods, local based and velocity based, generally computing x by minimizing a variational form or computing the mesh velocity $v = x_t$ using a Lagrangian like formulation. The moving mesh method we will employ belongs to the velocity based method, which is based on Liao's work (16; 17; 18; 19) and Moser's work (21). This method has several advantages: only linear Laplace problems on fixed mesh are needed to be solved, monitior function can be obtained directly from error distribution or distance function, mesh tangling can be prevented, and the data structure for mesh node and element system is always the same as that for the starting mesh.

Suppose $g(x)$ (area function) to be the area distribution on the undeformed mesh, while $f(x)$ (monitor function) in contrast describes the absolute mesh size distribution of the target grid, which is independent of the starting grid and chosen according to the need of physical problems. Then, the transformation φ can be computed via the following four steps:

1. Compute the scale factors c_f and c_g for the given monitor function $f(x) > 0$ and the area function g using

$$
c_f \int_\Omega \frac{1}{f(x)} dx = c_g \int_\Omega \frac{1}{g(x)} dx = |\Omega|, \tag{15}
$$

where Ω is a computational 2D domain, and $f(x) \approx$ local mesh area. Let $\tilde{f}$ and $\tilde{g}$ denote the reciprocals of the scaled functions f and g, i.e.

$$
\tilde{f} = \frac{c_f}{f}, \qquad \tilde{g} = \frac{c_g}{g} \tag{16}
$$

2. Compute a grid-velocity vector field $v : \Omega \to \mathbf{R}^n$ by satisfying the following linear Poisson equation

$$
\begin{aligned}
&-\mathrm{div}(v(x)) = \tilde{f}(x) - \tilde{g}(x), \quad x \in \Omega \\
&v(x) \cdot \mathbf{n} = 0, \quad x \in \partial\Omega
\end{aligned} \tag{17}
$$

where $\mathbf{n}$ being the outer normal vector of the domain boundary $\partial\Omega$, which may consist of several boundary components.

3. For each grid point x, solve the following ODE system

$$
\begin{aligned}
&\tfrac{\partial\varphi(x,t)}{\partial t} = \eta(\varphi(x,t),t), \quad 0 \le t \le 1 \\
&\varphi(x,0) = x
\end{aligned} \tag{18}
$$

with

$$
\eta(y,s) := \frac{v(y)}{s\tilde{f}(y) + (1-s)\tilde{g}(y)} \tag{19}
$$

where $y \in \Omega$ and $s \in [0,1]$

4. Get the new grid points via

$$
\varphi(x) := \varphi(x,1). \tag{20}
$$

ALE Formulation of the FBM

For a better approximation of the solid surfaces, we adopt the above described moving mesh method such that we can preserve the mesh topology as generalized tensorproduct or blockstructured meshes, while a local alignment with the rigid body surface is reached. The moving mesh method is sometimes referred to as the quasi-Lagrangian method. When the moving mesh method is applied to the multigrid FBM, a mesh velocity $\mathbf{W}_m$ in the convective term in Eq. (12b) should be introduced, i.e.,

$$
\rho_f \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \mathbf{W}_m) \cdot \nabla\mathbf{u} \right] - \nabla \cdot \sigma = 0 \tag{21}
$$

for $\mathbf{X} \in \Omega_f$. In the literature this is also referred to as an Arbitrary Lagrangian-Eulerian (ALE) formulation. Note that the mesh velocities $\mathbf{W}_m$ do not appear in the continuity equation, as a pressure-Poisson equation is solved to satisfy the continuity equation in an outer loop. Care has to be taken to satisfy the geometric conservation law (GCL), where the mesh velocity $\mathbf{W}_m$ must be equal to the movement of the mesh velocity $\Delta\mathbf{x}$ during the time step. Therefore, the mesh velocities $\mathbf{W}_m$ should be calculated according to the nodal movement from the previous time step by

$$\mathbf{W}_m = \frac{1}{\Delta t}(\mathbf{x}^{n+1} - \mathbf{x}^n) \tag{22}$$

where Δt is the time step size and n denotes the time step number. In each time step, a new deformation mesh is generated based on a starting equidistance mesh, then the system matrices are updated and the mesh velocity according to the new position of the deformation mesh nodes should be calculated.

Time Discretization by Fractional-Step-θ Scheme

The fractional-step-θ 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 Refs. (22; 24). We first semi-discretize the Eqs. (21) and (12) (a) in time by the fractional-step-θ 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-θ-scheme, one macro time step $t_n \to t_{n+1} = t_n + K$ is split into three consecutive substeps with $\tilde{\theta} := \alpha\theta K = \beta\theta' K$,

$$\begin{aligned}
&[I + \tilde{\theta} N(\mathbf{u}^{n+\theta})]\mathbf{u}^{n+\theta} + \theta K \nabla p^{n+\theta} = \\
&\qquad = [I - \beta\theta K N(\mathbf{u}^n)]\mathbf{u}^n \\
&\nabla\cdot \mathbf{u}^{n+\theta} = 0 \\
&[I + \tilde{\theta} N(\mathbf{u}^{n+1-\theta})]\mathbf{u}^{n+1-\theta} + \theta' K \nabla p^{n+1-\theta} = \\
&\qquad = [I - \alpha\theta' K N(\mathbf{u}^{n+\theta})]\mathbf{u}^{n+\theta} \\
&\nabla\cdot \mathbf{u}^{n+1-\theta} = 0 \\
&[I + \tilde{\theta} N(\mathbf{u}^{n+1})]\mathbf{u}^{n+1} + \theta K \nabla p^{n+1} = \\
&\qquad = [I - \beta\theta K N(\mathbf{u}^{n+1-\theta})]\mathbf{u}^{n+1-\theta} \\
&\nabla\cdot \mathbf{u}^{n+1} = 0
\end{aligned} \tag{23}$$

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

$$N(\mathbf{v})\mathbf{u} := -\nu\ \nabla\cdot\ \left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right] + (\mathbf{v} - \mathbf{W}_m)\cdot\nabla\mathbf{u} \tag{24}$$

Therefore, from Eq. (23) in each time step, we have to solve nonlinear problems of the following type,

$$\begin{aligned}
&[I + \theta_1 K N(\mathbf{u})]\mathbf{u} + \theta_2 K \nabla p = \mathbf{f} \\
&\mathbf{f} := [I - \theta_3 K N(\mathbf{u}^n)]\mathbf{u}^n \\
&\nabla\cdot\mathbf{u} = 0
\end{aligned} \tag{25}$$

For the Eq. (12) (c), we simply take an explicit expression like,

$$\mathbf{u}^{n+1} = \mathbf{U}_i^n + \omega_i^n \times (\mathbf{X}^n - \mathbf{X}_i^n)\,. \tag{26}$$

Space Discretization by Finite Element Method

If we define a pair $\{\mathbf{u}, p\} \in H := \mathbf{H}_0^1(\Omega) \times L := \mathrm{L}_0^2(\Omega)$, and two 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. (25) reads as follows:

$$\begin{cases}
(\mathbf{u}, \mathbf{v}) + \theta_1 K\,[\,a(\mathbf{u}, \mathbf{v}) + n(\mathbf{u}, \mathbf{u}, \mathbf{v})\,] + \\
\qquad + \theta_2 K\, b\,(p, \mathbf{v}) = (\mathbf{f}, \mathbf{v})\,, \quad \forall\,\mathbf{v} \in H \\
b\,(q, \mathbf{u}) = 0\,, \quad \forall\, q \in L
\end{cases} \tag{27}$$

Here $\mathrm{L}_0^2(\Omega)$ 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:

$$n(\mathbf{u}, \mathbf{v}, \mathbf{w}) := \int_\Omega [u_i - (w_m)_i] \left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right) w_j\, dx \tag{28}$$

To discretize the Eq. (27) in space, we introduce a regular finite-element quadrilateral T_h for the whole computational domain Ω_T, where h is the symbol used as a parameter characterizing the maximum width of the elements of T_h. To obtain the fine mesh T_h from a coarse mesh T_{2h}, we simply connect opposing midpoints. In the fine grid T_h, the old midpoints of the coarse mesh T_{2h} become vertices. We choose the $\tilde{Q}1/Q0$ element pair which uses rotated bilinear shape function for the velocity spanned by $\langle x^2 - y^2, x, y, 1\rangle$ in 2D and piecewise constants for the pressure in cells. The nodal values are the mean values of the velocity vector over the element edges, and the mean values of the pressure over the elements rendering this approach nonconforming. The nonconforming $\tilde{Q}1/Q0$ element pair has several important features. It satisfies the Babuška–Brezzi condition without any additional stabilization, and the stability constant seems to be independent of the shape and size of the element. In particular on meshes containing highly stretched and anisotropic cells, the stability and the approximation properties are always satisfied. In addition, it admits simple upwind strategies which lead to matrices with certain M-matrix properties (22). If we choose finite-dimensional spaces H_h and L_h and define a pair $\{\mathbf{u}_h, p_h\} \in H_h \times L_h$, the discrete problem of Eq. (27) reads:

$$\begin{cases}
(\mathbf{u}_h, \mathbf{v}_h) + \theta_1 K\,[\,a_h(\mathbf{u}_h, \mathbf{v}_h) + \tilde{n}_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h)\,] + \\
\qquad + \theta_2 K\, b_h(p_h, \mathbf{v}_h) = (\mathbf{f}, \mathbf{v}_h)\,, \quad \forall\,\mathbf{v}_h \in H_h \\
b_h(q_h, \mathbf{u}_h) = 0\,, \quad \forall\, q_h \in L_h
\end{cases} \tag{29}$$

where

$$\begin{aligned}
a_h(\mathbf{u}_h, \mathbf{v}_h) &:= \textstyle\sum_{T\in T_h} a(\mathbf{u}_h, \mathbf{v}_h)_{|T} \\
b_h(p_h, \mathbf{v}_h) &:= \textstyle\sum_{T\in T_h} b(p_h, \mathbf{v}_h)_{|T}.
\end{aligned} \tag{30}$$

Note that $\tilde{n}_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h)$ is a new convective term which includes streamline-diffusion stabilizations defined by:

$$\begin{aligned}\tilde{n}_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) &:= \textstyle\sum_{T\in T_h} n(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)_{|T} + \\ &+ \textstyle\sum_{T\in T_h} \delta_T(\mathbf{u}_h\cdot\nabla\mathbf{v}_h, \mathbf{u}_h\cdot\nabla\mathbf{w}_h)_{|T}\end{aligned} \qquad (31)$$

here δ_T is a local artificial viscosity which is a function of a local Reynolds number Re_T,

$$\delta_T := \delta^* \cdot \frac{h_T}{||\mathbf{u}||_\Omega} \cdot \frac{2Re_T}{1+Re_T}, \qquad Re_T = \frac{||\mathbf{u}||_T \cdot h_T}{\nu} \quad (32)$$

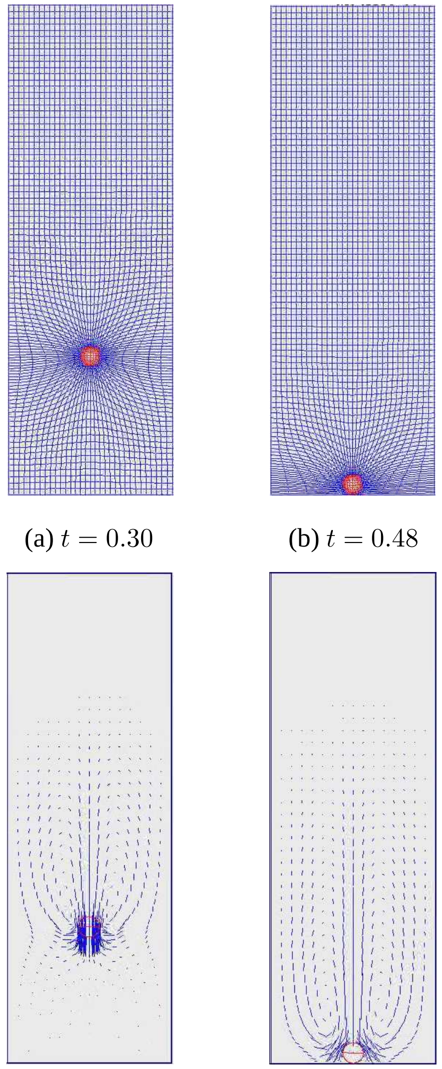
where $||\mathbf{u}||_\Omega$ means the maximum norm of velocity in Ω_T, $||\mathbf{u}||_T$ is an averaged norm of velocity over T, h_T denotes local mesh size of T, and δ^* is an additional free parameter which can be chosen arbitrarily ($\delta^* = 0.1$ is used in our calculations, also see (22)). Obviously, for small local Reynolds numbers, with $Re_T \to 0$, δ_T is decreasing such that we reach in the limit case the standard second order central discretization. Vice versa, for convection dominated flows with $Re_T >> 1$, we add an anisotropic diffusion term of size $O(h)$ which is aligned to the streamline direction $\mathbf{u}_h$.

Numerical Results

The correct simulation of the proposed method is verified in this section. The case of single moving particle in the fluid is presented to further validate the improvement of accuracy and efficiency through using the presented method. Finally, the drafting, kissing and tumbling of two disks in a channel and the agglomeration model applied for two particles are provided to show the presented method can be easily implemented in the simulation of particulate flows with large number of particles.

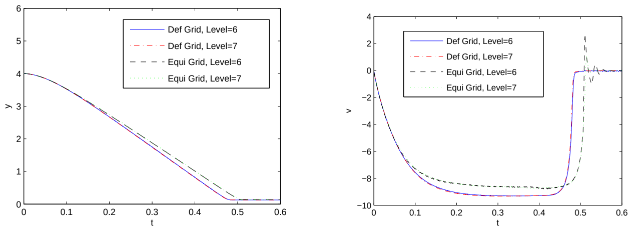
One 2D Circular Ball Falling Down in a Channel

Now we perform a real particle motion as follows: The computational domain is a channel of width 2 and height 6. A rigid circular ball with diameter $d = 0.25$ and density $\rho_p = 1.5$ 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 $\nu = 0.1$, the gravitational acceleration velocity $g = -980$. We suppose that the ball and the fluid are initially at rest. The simulation is carried out on fixed equidistance meshes and moving deformation meshes, respectively, each of them having two different level, i.e., $\text{Level} = 6$ with 12545 nodes and 12288 elements, as well as $\text{Level} = 7$ with 49665 nodes and 49152 elements. Fig. 1 gives two instantenous results at $t = 0.30$ and $t = 0.48$ of deformation mesh and vector field, respectively. Fig. 2 presents the comparison of the time history of y-coordinate and v-component velocity of the center of the ball by using equidistance meshes and deformation meshes, each of them is calculated by two level meshes $\text{LEVEL} = 6$ and $\text{LEVEL} = 7$, respectively. If we compare these results with those obtained by Glowinski in Ref. (8), we can find that the results on the deformation meshes are much closer to Glowinski's results than those of the equidistance meshes, which shows the expected higher accuracy and efficiency.


(a) $t = 0.30$ (b) $t = 0.48$

(c) $t = 0.30$ (d) $t = 0.48$

Figure 1: 2D circular ball falling down in a channel



(a) $y \sim t$ (b) $v \sim t$

Figure 2: The time history of y and v of the center of a 2D ball falling down in a channel

Drafting, Kissing and Tumbling of Two Disks in a Channel

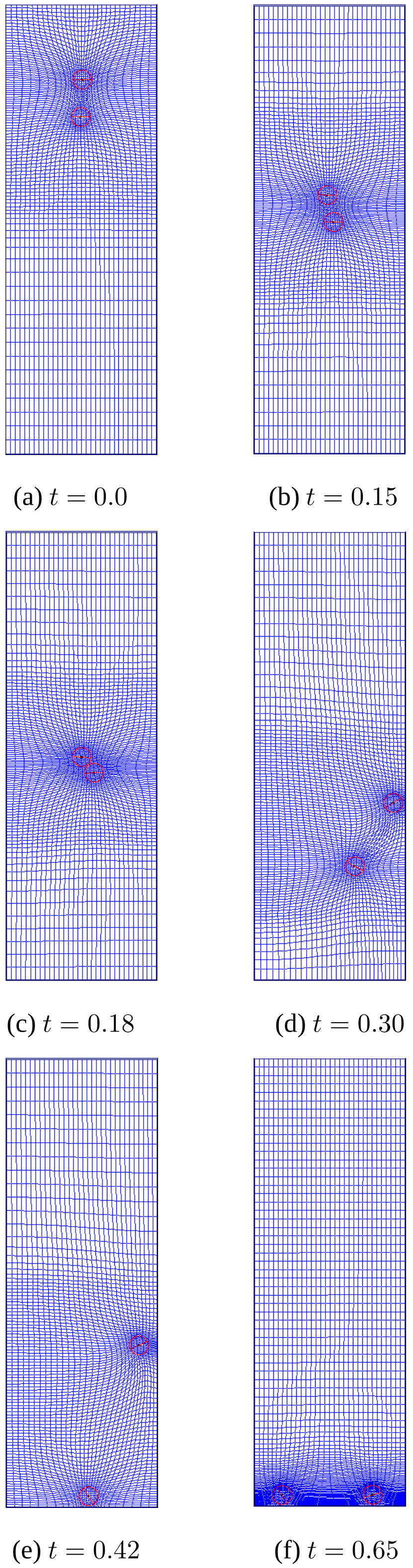
In the following two numerical experiments, we will carry on the cases of multiple particles in a fluid to show that the presented method can be easily implemented for the real particulate flows with many moving particles.

When two particles are dropped close to each other, they interact by undergoing "drafting, kissing and tumbling" (27), which is often chosen to examine the complete computational model of particulate flows, including the prevention of collisions. Therefore, we also study the sedimentation of two circular particles in a two-dimensional channel, comparing the results with respect to two different level of grid deformation mesh sizes and regarding the results in (8). The computational domain is a channel of width 2 and height 6. Two rigid circular disks with diameter $d = 0.25$ and density $\rho_p = 1.5$ are located at $(1, 5)$ (No.1 disk) and $(1, 4.5)$ (No.2 disk) at time $t = 0$, and they are falling down under gravity in an incompressible fluid with density $\rho_f = 1$ and viscosity $\nu = 0.01$, the gravitational acceleration velocity $g = -980$. We suppose that the disks and the fluid are initially at rest. The simulation is carried out on the moving deformation meshes, having two different level, i.e., $\text{Level} = 7$ with 49665 nodes and 49152 elements, as well as $\text{Level} = 8$ with 197633 nodes and 196608 elements.

Fig. 3 shows the moving deformation meshes employed in simulation of the two falling disks. The grid lines are always concentrated around the surfaces of the two disks and in the region of the gap between the two disks, and meantimes move with the two disks during the computations. From these figures, we can see that the disk in the wake (No.1 particle) falls more rapidly than the disk 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 $t = 0.15$. After touching, the two disks fall together until they tumble ($t = 0.18$) and subsequently they separate from each other ($t = 0.30$). The tumbling of the disks takes place because the configuration, when both are parallel to the flow direction, is unstable. The No.1 disk is touching first the bottom wall at $t = 0.42$, while the No.2 disk reaches the bottom wall at $t = 0.65$. All results compare qualitatively well with those presented in (6; 8; 28; 29).

Agglomeration results

As we have described earlier in this paper, here we provide an numerical result regarding the agglomeration model we have tested. We performed a similar simulation like the one for drafting, tumbling and kissing method using the agglomeration model, described earlier in the paper, instead of the collision model. So, we started with two circular disks, very close one to each other, in the middle part of a channel with width 2 and height 6. Fig. 4 shows how the particles behave. We can observe that first of all, the particles come very close one to each other, then they touch and further on they remain connected. For this simulation we used only the FBM method without moving mesh method, just to test if our agglomeration model is valid. Also is possible to simulate the same situation using the method describe in this paper: FBM with moving mesh method. However, the results presented here are dependent on the trick made by us over the collision model and future purpose for the research is to develop a much better model for the agglomeration case.


(a) $t = 0.0$

(b) $t = 0.15$

(c) $t = 0.18$

(d) $t = 0.30$

(e) $t = 0.42$

(f) $t = 0.65$

Figure 3: Deformation meshes for two circular disks falling down in a channel

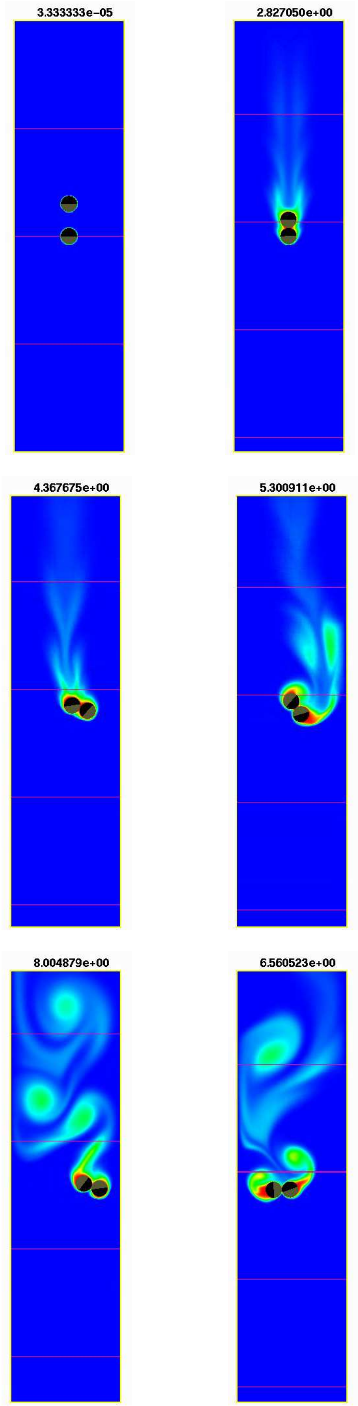


Figure 4: Agglomeration model for two circular disk falling down in a channel

Conclusions

We have presented the combination of the multigrid fictitious boundary method and a moving mesh method for the simulations of particulate flow with many moving rigid particles. The new approach directly improves the accuracy upon the previous pure multigrid FBM for particulate flows. It is also computationally cheap and simple to implement. Since the size of computation and data structure of the moving deformation meshes are fixed, this enables the proposed method much easier to incorporate into most CFD codes without the need for the changing of system matrix structures and special interpolation procedures. It is suitable to accurately and efficiently perform the direct numerical simulation of particulate flows with large number of moving particles. Numerical examples of single moving particle in a fluid as well as the drafting, kissing and tumbling of two disks in a channel have been presented to show that the presented method can significantly improve the accuracy for dealing with the interaction between the fluid and the particles, and can be easily applied to real particulate flows with many moving particles. Also, for a new direction of research we tested and presented an numerical example for two disks in a channel using the agglomeration model which can be also tested with the presented method and shows good and accurate results.

References

[1] Hu, H.H., Joseph, D.D., Crochet, M.J.: Direct Simulation of Fluid Particle Motions. Theor. Comp. Fluid Dyn., **3**, 285 – 306 (1992)

[2] Hu, H.H., Patankar, N.A., Zhu, M.Y.: Direct Numerical Simulations of Fluid-Solid Systems Using the Arbitrary Lagrangian-Eulerian Techniques. J. Comput. Phys., **169**, 427 – 462 (2001)

[3] Galdi, G.P., Heuveline, V.: Lift and Sedimentation of particles on the Flow of a Viscoelastic Liquid in a Channel. CRC Press, Free and Moving Boundaries: Analysis, Simulation and Control (2007)

[4] Maury, B.: Direct Simulations of 2D Fluid-Particle Flows in Biperiodic Domains. J. Comput. Phy., **156**, 325 – 351 (1999)

[5] Glowinski, R., Pan, T.W., Hesla, T.I., Joseph, D.D.: A Distributed Lagrange Multiplier/Fictitious Domain Method for Particulate Flows. Int. J. Multiphase Flow, **25**, 755 – 794 (1999)

[6] Patankar, N.A., Singh, P., Joseph, D.D., Glowinski, R., Pan, T.W.: A New Formulation of the Distributed Lagrange Multiplier/Fictitious Domain Method for Particulate Flows. Int. J. Multiphase Flow, **26**, 1509 – 1524 (2000)

[7] Glowinski, R., Pan, T.W., Hesla, T.I., Joseph, D.D., 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**, 363 – 426 (2001)

[8] Glowinski, R.: Finite Element Methods for Incompressible Viscous Flow. In *Handbook of numerical analysis*, **Vol. IX**, Ciarlet, P.G and Lions, J.L., Editors, North-Holland, Amsterdam, 701 – 769 (2003)

[9] Turek, S., Wan, D.C., 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, 37 – 68 (2003)

[10] Wan, D.C., Turek, S., Rivkind, L.S.: An Efficient Multigrid FEM Solution Technique for Incompressible Flow with Moving Rigid Bodies. Numerical Mathematics and Advanced Applications, ENUMATH 2003, Springer, 844 – 853 (2004)

[11] Wan, D.C., Turek, S.: Direct Numerical Simulation of Particulate Flow via Multigrid FEM Techniques and the Fictitious Boundary Method. Int. J. Numer. Method in Fluids, Vol. **5**, Nr. 51, 531 – 566 (2006)

[12] Wan, D.C., Turek, S.: An Efficient Multigrid-FEM Method for the Simulation of Liquid-Solid Two Phase Flows. J. for Comput. and Appl. Math., Vol **203**, Nr. 2, 561 – 580 (2007)

[13] Berger, M.J., Collela, P.: Local Adaptive Mesh Refinement for Shock Hydrodynamics. J. Comput. Phy., **82**, 64 – 84 (1989)

[14] Huang, W.: Practical Aspects of Formulation and Solution of Moving Mesh Partial Differential Equations. J. Comput. Phy., **171**, 753 – 775 (2001)

[15] Ceniceros, H.D., Hou, T.Y.: An Efficient Dynamically Adaptive Mesh for Potentially Singular Solutions. J. Comput. Phy., **172**, 609 – 639 (2001)

[16] Bochev, P.B., Liao, G., de la Pena, G.C.: Analysis and Computation of Adaptive Moving Grids by Deformation. Numerical Methods for Partial Differential Equations, **12**, 489 (1996)

[17] Liao, G., Semper, B.: A Moving Grid Finite–Element Method Using Grid Deformation. Numerical Methods for Partial Differential Equations, **11**, 603 – 615 (1995)

[18] Liu, F., Ji, S., Liao, G.: An Adaptive Grid Method and its Application to Steady Euler Flow Calculations. SIAM Journal on Scientific Computing, **20**(3), 811 – 825 (1998)

[19] Cai, X.X., Fleitas, D., Jiang, B., Liao, G.: Adaptive Grid Generation Based on Least–Squares Finite–Element Method. Computers and Mathematics with Applications, **48**(7-8), 1077 – 1086 (2004)

[20] Grajewski, M., Köster, M., Kilian, S., Turek, S.: Numerical Analysis and Practical Aspects of a Robust and Efficient Grid Deformation Method in the Finite Element Context. submitted to SISC (2005)

[21] Dacorogna, B., Moser, J.: On a Partial Differential Equation Involving the Jacobian Determinant. Annales de le Institut Henri Poincare, **7**, 1 – 26 (1990)

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

[23] Turek, S.: **FEATFLOW**–Finite element software for the incompressible Navier–Stokes equations: User Manual, Release 1.2, University of Dortmund (1999)

[24] 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, 987 – 1011 (1996)

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

[26] Schäfer, M., 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, 547 – 566 (1996)

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

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

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