FINITE ELEMENT FICTITIOUS BOUNDARY METHODS (FEM-FBM) FOR 3D PARTICULATE FLOW

R. Münster*, O. Mierka* and S. Turek*

 * Institute of Applied Mathematics (LS III), TU Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany
 e-mail: {raphael.muenster, omierka, stefan.turek}@math.uni-dortmund.de

Key words: 3D Particulate Flow, FEM, Multigrid, Fictitious Boundary, Grid Deformation

Abstract. In this paper we discuss numerical simulation techniques using a finite element approach in combination with the fictitious boundary method (FBM) for rigid particulate flow configurations in 3D. The flow is computed with a multigrid finite element solver (FEAT-FLOW), the solid particles are allowed to move freely through the computational mesh which can be static or adaptively aligned by a dynamic grid deformation method allowing structured as well as unstructured meshes. We explain the details of how we can use the fictitious boundary method to simulate flows with complex geometries, that are hard to describe analytically. Stationary and time-dependent numerical examples, demonstrating the use of such geometries are provided. Our numerical results include well-known benchmark configurations showing that the method can accurately and efficiently handle prototypical particulate flow situations in 3D with particles of different shape and size.

1 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 sedimenting and fluidized suspensions, lubricated transport, hydraulic fracturing of reservoirs, slurries, understanding solid-liquid interaction or medical applications. Several numerical simulation techniques for particulate flow have been developed over the past two decades. 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 is resolved so that the hydrodynamic force between the particle and the fluid can be explicitly obtained. The origins of finite element based simulations of particulate flows in Newtonian and viscoelastic fluids can be found in the work of Hu, Joseph and coworkers [18, 19], Galdi [10] and Maury [22]. This approach is based on the use of unstructured grids and an Arbitrary Lagrangian-Eulerian

(ALE) technique. Both the fluid and solid equations of motion are incorporated into a single coupled variational equation. At each time step, a new mesh is generated when the old one becomes too distorted, and the flow field is projected onto 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.

A different approach is based on the concept of 'ficitious domains'. The idea of this Eulerian method is to simulate the presence of an additional domain, the 'fictitious domain', inside the computational domain. This in turn implies the existence of a boundary between these domains, this boundary is realized by adding appropriate terms to the mathematical model. Applying this concept to particulate flows, the particle domain is treated as a fluid with additional constraints to impose proper rigid body motion. Several implementational variants of this general principle exist, the variants differ mainly in the mathematical realization of the boundary, the way the rigid motion is imposed on the particle domain or in the numerical solution process of the problem. Apart from these differences, all the methods have in common that they allow the use of a fixed grid, not depending on remeshing. The idea was first published by Glowinski, Joseph, Patankar and coauthors [11, 26, 12, 13], who proposed a approach based on 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 the particle domain is constrained to rigid body motion. The fluid-particle coupling is treated implicitly using a combined weak formulation in which the mutual forces cancel. Pan et al. [27] applied the method in a 3D simulation with 1204 spheres. The following work by Patankar and Sharma [28, 29] further enhanced the efficiency and decreased the solution time, which was achieved by an efficient projection scheme and a fast implementation of the rigid motion constraint. Further recent publications of 3D particulate flow simulations using the fictitious domain method include Boenisch [3], Cottet/Coquerelle [6] and Blasco [2]. Our group [34, 35, 36] presented another multigrid fictitious boundary method (FBM) for the detailed simulation of particulate flows which is based on a fixed unstructured FEM background grid. The motion of the solid particles is modeled 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. An advantage 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 particulate flows, yet the quest for more accurate and efficient methods remains active. An underlying problem when adopting the fictitious domain methods is that the boundary approximation is of low accuracy only. Particularly 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 is to preserve the mesh topology, for instance as generalized tensorproduct or blockstructured meshes, while

a local alignment with the physical boundary of the solid particles is achieved by a moving mesh process, such that the boundary approximation error can be significantly decreased.

The primary objective of this paper is to give a formulation of the multigrid fictitious boundary method (FBM) in 3D, to describe the details of how complex geometry can be used with the FBM and to show the validity of the methodology applying it to benchmark configurations. In these benchmarks, we will use the FBM to simulate an incompressible flow with stationary and moving rigid bodies. In this context our focus lies on the calculation of the forces acting on the rigid bodies, which are highly important for the study of the interaction between the fluid and the rigid bodies. In particular, these forces are the drag, lift and torque forces, which act on the surface of the rigid body. This gives rise to some problems, that have to be overcome in order to effectively use the FBM. These problems are that the surface of the rigid body is implicitly represented by a fixed mesh, so the quality of the surface approximation and consequently the accuracy of the drag/lift calculations depends on the refinement level of the grid. Furthermore, the surface is represented implicitly, so the surface integral formulation of the drag and lift forces cannot be used directly in the FBM framework. In the 2D case our remedy for the latter problem was to integrate over the area occupied by the rigid body rather than integrating over the boundary of the rigid body. In the 3D case this generalizes to a volume integral formulation. The remaining problem of surface approximation we intend to solve by a grid deformation method [16], which was shown to significantly improve results in the 2D case [36]. Another important topic concerning particulate flows is the treatment of collisions, which can be a highly difficult task in case of multiple moving rigid bodies of complex shape in 3D. For our current work we use the collision model presented in [34], which was extended to 3D and non-spherical geometries. In future work we consider using more sophisticated collision models based on the work of Maury [23], Ardekani/Rangel [1] or Lefebvre [20]. The organization of the paper is as follows: In Section 2, the physical models, our technique for complex geometries together with collision and agglomeration models for rigid particulate flows are described. The detailed numerical schemes including the multigrid FBM method are given in Section 3. Numerical experiments are implemented and their computational results will be presented in Section 4. The concluding remarks will be given in Section 5. In the appendix we present stationary results of the multigrid FBM combined with grid deformation.

2 DESCRIPTIONS OF THE PHYSICAL MODELS

2.1 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, ..., 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 fluid at time t, and $\Omega_i(t)$ as the domain occupied by the fluid at time t, and $\Omega_i(t)$ as the domain occupied by the fluid at time t, and $\Omega_i(t)$ as the domain occupied by the following Navier-Stokes equations in $\Omega_f(t)$,

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

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

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

Here, I is the identity tensor, $\mu_f = \rho_f \cdot \nu$, p is the pressure and 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 Newton-Euler equations, i.e., the translational velocities \mathbf{U}_i and angular velocities $\omega_i = (\omega_x, \omega_y, \omega_z)$ of the *i*th particle satisfy

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

where M_i is the mass of the *i*th particle; \mathbf{I}_i is the moment of inertia tensor of the *i*th particle about its center of mass; 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 *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 subsection. \mathbf{F}_i and \mathbf{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

$$\mathbf{F}_{i} = (-1) \int_{\partial \Omega_{i}} \boldsymbol{\sigma} \cdot \mathbf{n} \, d \, \Gamma_{i} \,, \qquad \mathbf{T}_{i} = (-1) \int_{\partial \Omega_{i}} (\mathbf{X} - \mathbf{X}_{i}) \times (\boldsymbol{\sigma} \cdot \mathbf{n}) \, d \, \Gamma_{i}, \tag{4}$$

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

$$\frac{d \mathbf{X}_i}{d t} = \mathbf{U}_i. \tag{5}$$

The angular velocity ω_i and the angle $\theta_i = (\theta_x, \theta_y, \theta_z)$ are calculated from the angular acceleration a_i and torque \mathbf{T}_i of the *i*th particle using the following equations:

$$\mathbf{T}_i = \mathbf{I}_i a_i \tag{6}$$

$$\frac{d\omega_i}{dt} = a_i \tag{7}$$

$$\frac{d\theta_i}{dt} = \omega_i \tag{8}$$

We plug Eq. (7) into Eq. (6), then multiply by the moment of inertia tensor \mathbf{I}_i^{-1} and integrate the torque to get the angular velocity ω_i . The angle θ_i can hence be calculated by integrating the angular velocity. No-slip boundary conditions are applied at the interface $\partial \Omega_i$ between the *i*th particle and the fluid, i.e., for any $\mathbf{X} \in \overline{\Omega}_i$, the velocity $\mathbf{u}(\mathbf{X})$ is defined by

$$\mathbf{u}(\mathbf{X}) = \mathbf{U}_i + \omega_i \times (\mathbf{X} - \mathbf{X}_i).$$
(9)

2.2 Collision and Agglomeration Models

Following existing modeling approaches, we examine a special collision model with a new definition of short range repulsive forces which can not only prevent the particles from getting too close, it 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 corresponding repulsive force is determined as

$$\mathbf{F}_{i,j}^{P} = \begin{cases} 0, & \text{for } d_{i,j} > R_i + R_j + \rho, \\ \frac{1}{\epsilon'_{P}} (\mathbf{X}_i - \mathbf{X}_j) (R_i + R_j - d_{i,j}), & \text{for } d_{i,j} < R_i + R_j, \\ \frac{1}{\epsilon_{P}} (\mathbf{X}_i - \mathbf{X}_j) (R_i + R_j + \rho - d_{i,j})^2, & \text{for } R_i + R_j \le d_{i,j} \le R_i + R_j + \rho, \end{cases}$$
(10)

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 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 *i*th particle, ρ_f is the fluid density), then we can take $\epsilon_P \simeq (\Delta h)^2$ and $\epsilon'_P \simeq \Delta h$ in the calculations. For the particle-wall collisions, the corresponding repulsive force reads

$$\mathbf{F}_{i}^{W} = \begin{cases} 0, & \text{for } d_{i}' > 2R_{i} + \rho, \\ \frac{1}{\epsilon'_{W}} (\mathbf{X}_{i} - \mathbf{X}_{i}')(2R_{i} - d_{i}'), & \text{for } d_{i}' < 2R_{i}, \\ \frac{1}{\epsilon_{W}} (\mathbf{X}_{i} - \mathbf{X}_{i}')(2R_{i} + \rho - d_{i}')^{2}, & \text{for } 2R_{i} \le d_{i}' \le 2R_{i} + \rho, \end{cases}$$
(11)

where \mathbf{X}'_i is the coordinate vector of the center of the nearest imaginary particle P'_i located on the boundary wall Γ w.r.t. the *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 , 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 exerted on the *i*th particle by the other particles and the walls can be expressed as follows,

$$\mathbf{F}'_{i} = \sum_{j=1, j \neq i}^{N} \mathbf{F}_{i,j}^{P} + \mathbf{F}_{i}^{W}.$$
(12)

Future plans for this research include also the development of an agglomeration model. As a first step in this direction, we perform a 'trick' to the described collision model such

that we obtain a prototypical agglomeration model. In Eq.(10) we multiply by -1 for $R_i + R_j \le d_{i,j} \le R_i + R_j + \rho$, in this manner the repulsive forces become attractive forces. The result is that the particles will not separate when they touch each other, but they will stick together. At the end of this paper, we will provide preliminary results for this simple agglomeration model while the development of more sophisticated agglomeration models is part of current research.

3 NUMERICAL METHOD

3.1 Multigrid FEM Fictitious Boundary Method

The multigrid FEM fictitious boundary method (FBM) [33, 30, 31] 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, ..., N), and it employs a fictitious boundary indicator (see [33]) which sufficiently describes all fine-scale structures of the particles with regard to the fluid-particle matching conditions of Eq. (9). 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 [30]). Hence, by making use of Eq. (9), 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 **u** is extended according to Eq. (9), which can be seen as an additional constraint to the Navier-Stokes equations (1), i.e.,

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 & (a) & \text{for } \mathbf{X} \in \Omega_T, \\ \rho_f \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \sigma = 0 & (b) & \text{for } \mathbf{X} \in \Omega_f, \\ \mathbf{u}(\mathbf{X}) = \mathbf{U}_i + \omega_i \times (\mathbf{X} - \mathbf{X}_i) & (c) & \text{for } \mathbf{X} \in \bar{\Omega}_i, \ i = 1, \dots, N. \end{cases}$$
(13)

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 surface integrals over the surfaces of the particles appear in the calculation of the forces \mathbf{F}_i and T_i . However, in the presented multigrid FBM method, the shapes of the surfaces of the moving particles are implicitly imposed in the fluid field. If we reconstruct the shapes of the 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. In order to resolve the shape, 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 ,

$$\alpha_i(\mathbf{X}) = \begin{cases} 1 & \text{for} \quad \mathbf{X} \in \Omega_i, \\ 0 & \text{for} \quad \mathbf{X} \in \Omega_T \setminus \Omega_i \end{cases}$$
(14)

where 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 surface of the *i*th particle, and approximates to the normal vector \mathbf{n}_i of surface of the *i*th particle defined on the grid, i.e., $\mathbf{n}_i \approx \nabla \alpha_i$. Then, the hydrodynamic forces acting on the *i*th particle can be computed by

$$\mathbf{F}_{i} = -\int_{\Omega_{T}} \boldsymbol{\sigma} \cdot \nabla \alpha_{i} \, d\,\Omega, \qquad T_{i} = -\int_{\Omega_{T}} \left(\mathbf{X} - \mathbf{X}_{i}\right) \times \left(\boldsymbol{\sigma} \cdot \nabla \alpha_{i}\right) \, d\,\Omega \,. \tag{15}$$

The integral over each element covering the whole domain Ω_T can be exactly calculated with a standard Gaussian quadrature of sufficiently high order. Since the gradient $\nabla \alpha_i$ is nonzero only near the 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.

3.2 Representation of Complex Geometry

Traditionally the geometry used in particulate flow simulations is restricted to shapes that can easily be described analytically. We decided to part with this restriction and to allow general complex shapes, for that an analytical description is not easily available. This is accomplished by representing the geometry as a surface triangulation, which is basically a piecewise linear approximation of the geometry's surface (see Fig. 1). This type of geometry is widely used in CAD, CAGD and computer graphics in general. In preliminary work our group developed efficient techniques for solving the geometric problems related to the use of such geometry in a FEM-FBM framework and analyzed the error of the geometry discretization [25, 24, 17]. The result of this work showed, that the geometry discretization error has no significant influence on the error of the numerical solution of the fluid-solid flow as a whole, because the error due to the spatial discretization of the domain is significantly larger than the error of the geometry discretization. Furthermore, we address how to calculate the function α_i in Eq. (15) for a geometry given as a surface triangulation. In terms of our numerical scheme this task can be rephrased as: how can we acquire the volume information needed to calculate Eq. (15) from the surface representation. In order to solve this task, we developed an algorithm based on the Jordan curve theorem. The Jordan curve theorem basically states (for the sake of simplicity we assume the 2D case), that a closed Jordan curve divides the plane into two distinct regions, the exterior and the interior region. From this we can follow that every ray, whose origin is in the interior region of the curve has an uneven number of intersections with the curve. This observation is the basis of our algorithm to compute the volume information we need. So we calculate the number of intersections of the ray with the complex geometry and based on this number, we decide whether a point is located in the interior region or the exterior region. This task can eventually be computationally expensive, so we additionally incorporate efficient hierarchical structures like bounding volume hierarchies to significantly reduce computation



(a) Surface triangulation of an el- (b) Complex non-convex object lipse

Figure 1: Examples of surface triangulations

time [25]. Our choice of geometry representation demands additional adjustements of the calculation of some physical parameters of the particles, in particular these parameters are the moment of inertia tensor and the mass centre of the particle. The moment of inertia tensor is a diagonal matrix, if the geometry is symmetric about the axes of rotation:

$$\left(\begin{array}{ccc} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{array}\right)$$

Here I_{xx} , I_{yy} and I_{zz} are the principal moments of inertia. If the geometry is not symmetric about all principal axes the tensor will have non-zero entries on the off-diagonals

$$\left(\begin{array}{ccc}I_{xx} & -I_{xy} & -I_{xz}\\-I_{xy} & I_{yy} & -I_{yz}\\-I_{xz} & -I_{yz} & I_{zz}\end{array}\right)$$

The off-diagonal entries are called the products of inertia. The principal moments are calculated by:

$$I_{xx} = \int_{P} \left(y^2 + z^2 \right) \delta\left(x, y, z \right) dP$$
(16)

$$I_{yy} = \int_{P} \left(x^2 + z^2 \right) \delta\left(x, y, z \right) dP$$
(17)

$$I_{zz} = \int_{P} \left(x^2 + y^2 \right) \delta\left(x, y, z \right) dP$$
(18)

Here $\delta(x, y, z)$ is the density distribution of the particle. The products of inertia are calculated likewise:

$$I_{xy} = \int_{P} xy\delta(x, y, z) dP$$
(19)

$$I_{xz} = \int_{P} xz\delta(x, y, z) dP$$
⁽²⁰⁾

$$I_{yz} = \int_{P} yz\delta(x, y, z) dP$$
(21)

In practice, we calculate these integrals using the same 'alpha technique' (see Eq. 15), that we employ in the volume integration for the drag and lift forces.

The algorithm of the multigrid FEM fictitious boundary method for solving the coupled system of fluid and particles can be summarized as follows:

- Given the positions and velocities of the particles, solve the fluid equations Eqs. (13)

 (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. (15). In the case of complex geometries apply the techniques described in this section and compute the collision forces by Eq. (12).
- 3. Solve Eq. (3) to get the translational and angular velocities of the particles to obtain the new positions and velocities of the particles by Eq. (3).
- 4. Use Eq. (13) (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.

3.3 Time Discretization by Fractional-Step- θ Scheme

The fractional-step- θ scheme is a strongly A-stable time stepping approach which possesses the full smoothing property being 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 nonenforced temporal oscillations. A more detailed discussion of these aspects can be found in [37, 38]. We first semi-discretize the Eqs. (13) (a) and (13) (b) 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 $\tilde{\theta} := \alpha \theta K = \beta \theta' K$,

$$[I + \tilde{\theta}N(\mathbf{u}^{n+\theta})]\mathbf{u}^{n+\theta} + \theta K \nabla p^{n+\theta} = [I - \beta \theta K N(\mathbf{u}^n)]\mathbf{u}^n$$

$$\nabla \cdot \mathbf{u}^{n+\theta} = 0,$$

$$[I + \tilde{\theta}N(\mathbf{u}^{n+1-\theta})]\mathbf{u}^{n+1-\theta} + \theta' K \nabla p^{n+1-\theta} = [I - \alpha \theta' K N(\mathbf{u}^{n+\theta})]\mathbf{u}^{n+\theta}$$

$$\nabla \cdot \mathbf{u}^{n+1-\theta} = 0,$$

$$[I + \tilde{\theta}N(\mathbf{u}^{n+1})]\mathbf{u}^{n+1} + \theta K \nabla p^{n+1} = [I - \beta \theta K N(\mathbf{u}^{n+1-\theta})]\mathbf{u}^{n+1-\theta}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0,$$
(22)

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,

$$N(\mathbf{v})\mathbf{u} := -\nu \,\nabla \cdot \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right] + \mathbf{v} \cdot \nabla \mathbf{u} \,. \tag{23}$$

Therefore, from Eq. (22), in each time step we have to solve nonlinear problems of the following type,

$$[I + \theta_1 K N(\mathbf{u})]\mathbf{u} + \theta_2 K \nabla p = \mathbf{f}, \quad \mathbf{f} := [I - \theta_3 K N(\mathbf{u}^n)]\mathbf{u}^n, \quad \nabla \cdot \mathbf{u} = 0.$$
(24)

For the Eq. (13) (c), we simply take an explicit expression, that means

$$\mathbf{u}^{n+1} = \mathbf{U}_i^n + \omega_i^n \times \left(\mathbf{X}^n - \mathbf{X}_i^n\right).$$
(25)

3.4 Space Discretization by Finite Element Method

If we define a pair $\{\mathbf{u}, p\} \in H := \mathbf{H}_0^1(\Omega) \times L := \mathbf{L}_0^2(\Omega)$, 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. (24) reads as follows:

$$\begin{cases} (\mathbf{u}, \mathbf{v}) + \theta_1 K [a(\mathbf{u}, \mathbf{v}) + n(\mathbf{u}, \mathbf{u}, \mathbf{v})] + \theta_2 K b(p, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) & \forall \mathbf{v} \in H \\ b(q, \mathbf{u}) = 0 & \forall q \in L \end{cases}$$
(26)

Here, $L_0^2(\Omega)$ and $H_0^1(\Omega)$ are the usual Lebesgue and Sobolev spaces, $n(\mathbf{u}, \mathbf{u}, \mathbf{v})$ is a trilinear form defined by

$$n(\mathbf{u}, \mathbf{v}, \mathbf{w}) := \int_{\Omega} [u_i - (W_m)_i] \left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right) w_j \, dx \,.$$
(27)

To discretize Eq. (26) in space, we introduce a regular finite element mesh T_h consisting of hexahedrons to cover the whole computational domain, where h characterizes the maximum edge length of the elements of T_h . To obtain the fine mesh T_h from a coarse mesh T_{2h} , we simply apply a regular refinement to the hexahedral cells that splits each hexahedron into 8 new hexahedrons. We choose the $\tilde{Q}1/Q0$ element pair which uses rotated bilinear shape function for the velocity spanned by $\langle x^2 - y^2, x^2 - z^2, x, y, z, 1 \rangle$ in 3D 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. 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 is 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 [37].

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. (26) reads,

$$\begin{cases} (\mathbf{u}_h, \mathbf{v}_h) + \theta_1 K \left[a_h(\mathbf{u}_h, \mathbf{v}_h) + \tilde{n}_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) \right] \\ + \theta_2 K b_h(p_h, \mathbf{v}_h) = (\mathbf{f}, \mathbf{v}_h), & \forall \mathbf{v}_h \in H_h \\ b_h(q_h, \mathbf{u}_h) = 0, & \forall q_h \in L_h \end{cases}$$
(28)

where $a_h(\mathbf{u}_h, \mathbf{v}_h) := \sum_{T \in T_h} a(\mathbf{u}_h, \mathbf{v}_h)_{|T}$ and $b_h(p_h, \mathbf{v}_h) := \sum_{T \in T_h} b(p_h, \mathbf{v}_h)_{|T}$. 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

$$\tilde{n}_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_{T \in T_h} n(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)_{|T} + \sum_{T \in T_h} \delta_T(\mathbf{u}_h \cdot \nabla \mathbf{v}_h, \mathbf{u}_h \cdot \nabla \mathbf{w}_h)_{|T} .$$
(29)

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

$$\delta_T := \delta^* \cdot \frac{h_T}{||\mathbf{u}||_{\Omega}} \cdot \frac{2Re_T}{1 + Re_T}, \qquad Re_T = \frac{||\mathbf{u}||_T \cdot h_T}{\nu}, \tag{30}$$

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 [37]). Obviously, for small local Reynolds numbers, with $Re_T \rightarrow 0$, 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 .

4 NUMERICAL RESULTS

As our first numerical example, we examine the case of a single sedimenting particle in a fluid to validate the accuracy and efficiency of the FEM-FBM method. As a second example, we demonstrate the drafting, kissing and tumbling phenomenon involving two balls in a channel. Further results of our collision and agglomeration models are provided to show that the presented method can easily be applied to particulate flows with several particles. Finally, we show examples of flows with complex objects. The first example is a sedimenting particle of a 'bean-like' shape, represented as a surface triangulation. Then, in the appendix of this paper we present preliminary results of a combination of the FEM-FBM with complex objects and grid deformation in the stationary case.

4.1 Numerical Simulation of a Freely Sedimenting Sphere in a Channel

In this section, we consider the the numerical simulation of a sphere freely falling in a tube with a circular cross-section. The configuration of this simulation is based on a well documented experiment [5]. We will calculate the terminal velocities of the particle with the FEM-FBM, this allows us to validate our method by comparing these values with the experimental data. The computational domain in this case is a channel Ω with a circular cross-section and the dimensions $\Omega = (0,1) \times (0,1) \times (0,2.5)$. A ball is initially placed at (0.5,0.5,0.5) and is allowed to fall freely under gravitiy. For this case, we use the infinite reference frame formulation of the Navier-Stokes equations, thus the channel has basically infinite length. Initially, the fluid velocity and the velocity of the ball are zero. The fluid density is $\rho_f = 1.0$. The viscosity of the fluid takes the values $\nu_f = 0.2, 0.1, 0.05, 0.02, 0.01$.



(a) Coarse grid of the falling (b) Flow field of the falling sphere sphere benchmark configuration

Figure 2: Configuration and flow field of the benchmark

The density of the ball is either $\rho_s = 1.02$ or 1.14, the diameter d of the ball can take the values 0.2 and 0.3. The time step for this simulation is $\Delta t = 0.0005$. In the simulation we used a mesh with 671.744 elements at level 4. The coarse grid of the benchmark (1312 elements) is shown in Fig. 4.1. As the results in table 1 - table 4 show, our computed terminal velocites are in good agreement with the experimental values [5]. For any configuration, the relative error of our numerical solution is less than 10 percent. Our results are comparable with Glowinski's and Pan's results [13], except for the fact that our method does not produce a significantly higher error in the case of $\nu = 0.01$. In fact, our code produces accurate results even in this case. The results in table 5 show the validity of the multigrid framework. The underlying assumption is that the computed results should approximate the reference values better on a higher resolution mesh. So for fixed values of viscosity ν , particle density ρ_s and particle diameter d we calculated the terminal velocity on different refinement levels. As expected we get better results as the refinement level increases.

ν	$U_{featflow}$	U_{exp}	Relative error $(\%)$
0.01	1.4660	1.4110	3.90
0.02	0.9998	0.9129	9.52
0.05	0.4917	0.4603	6.82
0.1	0.2637	0.2571	2.57
0.2	0.1335	0.1317	1.37

Table 1: Comparison of terminal velocities in simulation and experiment ($\rho_s = 1.02 \ d = 0.2$)

4.2 Drafting, Kissing and Tumbling of Two Spheres in a Channel

In the following two numerical experiments, we will analyze the case of several moving particles in a fluid. When two particles are dropped closely to each other, they interact by undergoing "drafting, kissing and tumbling" [9], which is often chosen to examine the complete

ν	$U_{featflow}$	U_{exp}	Relative error $(\%)$
0.01	2.167	2.107	2.84
0.02	1.495	1.436	4.11
0.05	0.809	0.749	8.01
0.1	0.402	0.404	0.44
0.2	0.218	0.216	1.02

Table 2: Comparison of terminal velocities in simulation and experiment ($\rho_s = 1.02 \ d = 0.3$)

Table 3: Comparison of terminal velocities in simulation and experiment ($\rho_s = 1.14 \ d = 0.2$)

ν	$U_{featflow}$	U_{exp}	Relative error $(\%)$
0.02	4.370	4.334	0.83
0.05	2.699	2.489	8.44
0.1	1.649	1.552	6.25
0.2	0.946	0.870	8.74

Table 4: Comparison of terminal velocities in simulation and experiment ($\rho_s = 1.14 \ d = 0.3$)

ν	$U_{featflow}$	U_{exp}	Relative error $(\%)$
0.02	5.885	6.283	6.33
0.05	4.133	3.972	4.05
0.1	2.588	2.426	6.66
0.2	1.492	1.401	6.50

Table 5: Comparison of terminal velocities for different refinement levels ($\nu = 0.02 \ \rho_s = 1.02 \ d = 0.2$)

LVL	$U_{featflow}$	U_{exp}	Relative error $(\%)$
3	1.0820	0.9129	18.64
4	0.9998	0.9129	9.52
5	0.9601	0.9129	5.17

computational model of particulate flows, including the prevention of collisions. Therefore, we also study the sedimentation of two spherical particles in a three-dimensional channel. The computational domain is a channel of dimension $\Omega = (0,2) \times (0,1) \times (0,2)$. Two rigid spheres with diameter d = 0.25 and density $\rho_p = 1.5$ are located at (1,0.5,1) (sphere No.1) and (1,0.5,1.5) (sphere No.2) 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 timestep size is $\Delta t = 0.005$. We suppose that the spheres and the fluid are initially at rest. The simulation is carried out on a regular mesh at refinement level 7 with 274625 nodes and 262144 elements. Fig. 3 . From these figures, we can see that the particle in the wake (No.1 particle) falls more rapidly than the particle No.2 in front since the fluid forces acting on it are smaller. The gap

between them decreases, and they almost touch ("kiss") each other at time t = 0.185. After touching, the two particles fall together until they tumble (t = 0.235) and subsequently they separate from each other (t = 0.285). The tumbling of the particles takes place because the configuration, when both are parallel to the flow direction, is unstable.

4.3 Agglomeration results

As we have described earlier, we also provide numerical results regarding the proposed simple agglomeration model. We use the following domain $\Omega = (0,2) \times (0,2) \times (0,2)$ with 260.644 cells and apply an infinite channel technique. In this simulation the following fluid parameters are used: density $\rho_f = 1$ and viscosity $\nu = 0.001$. The timestep size is $\Delta t = 0.005$. We place two spheres (d = 0.25, $\rho = 2.5$) with density $\rho_p = 1.5$ very close one to each other, in the middle part of the channel at (1.0, 1.0, 1.0) and (1.3, 1.0, 1.0). An initial particle velocity of $u_p 1=0.2$ and $u_p 2=-0.2$ is prescribed to ensure that the particles quickly collide. Figures. 4(a)-4(f) show the behaviour of the particles and illustrate the forces acting on the particles. Futhermore, Fig. 4 displays for each timestep the velocity field, the agglomeration forces (blue) and the total velocity of each particle (green). We can observe, that the particles during t = 0 and t = 0.075 quickly come very close to each other, but the agglomeration forces are inactive, as intended. Then, in timestep t = 0.075 they are so close, that the agglomeration forces start to act. They move and rotate according to the flow field, while the agglomeration forces keep them connected. While this model fullfills the task of keeping particles connected, we intend to incorporate much more sophisticated models. In particular we favor the model proposed by Lefebvre [20], a variant of which we want to integrate into our particulate flow framework.

4.4 Sedimentation of a complex particle

Our last example is a particulate flow configuration with a complex geometry represented by a surface triangulation using the techniques described in section 3.2. We use the 'beanshaped' particle shown in Fig. 1(b) for this simulation. The computational domain $\Omega =$ $(0, 2.2) \times (0, 0.41) \times (0, 0.41)$ is covered by a grid with 81.920 elements at refinement level 4, as in the preceeding examples a infinite channel technique is applied. This technique allows us to use a higher resolution around the particle and in its wake, while using a lower resolution in regions that have only little influence on the particle motion. Figures 5 and 6 show how the particle freely sediments according to the computed velocity field. The depicted sequence shows the evolution of the particle from t = 3.8 to t = 4.5 in steps of $\Delta t = 0.1$, the timeline proceeds from left to right and from top to bottom.

5 Conclusions

We have presented a 3D FBM-FEM technique for the direct simulation of moving boundaries. The method has been tested in various prototypical particulate flow configurations. We showed how complex geometry can easily be used in the FBM-FEM context. In the single particle case our method proved to yield accurate results in the terminal velocity benchmark configurations. In these examples, we did not observe a loss of significant loss of accurracy as was observed in other codes. The numerical example of drafting, kissing and tumbling shows, that our method is able to model the interaction between the particles and the fluid and to reproduce the expected behaviour. Our prototypical study of particle agglomeration shows that these phenomena can easily and efficiently be included into the FBM-FEM framework. Thus, the extension to a more realistic agglomeration model does not require significant changes in the numerical solution scheme. The next steps in improving our method focus on improving the accuracy and numerical efficiency.



Figure 3: Drafting, Kissing and Tumbling for t = 0.05, t = 0.15, t = 0.185, t = 0.235, t = 0.285, t = 0.4, t = 0.56, column: front view, right column: side view



(f) Particles at t = 0.7

Figure 4: Behaviour of two particles with agglomeration forces. Left column: particles and the flow field, center column: blue vectors at the particles show the agglomeration forces only, right column: green vectors on the particles show the total velocity of the particles



Figure 5: Particulate flow with a complex geometry from t=3.8 to t=4.5. Time advancement from left to right and from top to bottom



Figure 6: Sedimentation of a complex particle t=3.8 to t=4.8 from a different angle of view.

6 Appendix: Grid Deformation

In this section we want to provide further evidence for our assumptions that adaptive grid deformation can improve the accuracy of computation results and geometry approximation in particulate flow simulations. We do this by presenting results of stationary simulations using the grid deformation method described in the following section.

6.1 Grid Deformation Algorithm

We briefly describe the grid deformation method which we combined with our stationary Multigrid-FBM Navier-Stokes solver. The details of the grid deformation method can be found in Ref. [15]. The moving mesh problem can be equated to constructing a transformation φ , $x = \varphi(\xi)$ from computational space (with coordinate ξ) to physical space (with coordinate x). There are two basic types of grid deformation methods, generally computing x by minimizing a variational form or computing the mesh velocity $v = x_t$ using a Lagrangian like formulation. The applied grid deformation method belongs to the velocity based methods, which is based on Liao's [4, 39, 21, 7] and Moser's work [8]. This method has several advantages: only linear Poisson problems on fixed meshes are needed to be solved, monitor functions can be obtained directly from error distributions or distance functions, mesh tangling can be prevented, and the data structure for the mesh nodes is always the same as for the starting mesh.

Suppose g(x) (area function) to be the area distribution on the undeformed mesh, while f(x) (monitor function) 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

$$c_f \int_{\Omega} \frac{1}{f(x)} dx = c_g \int_{\Omega} \frac{1}{g(x)} dx = |\Omega|,$$
(31)

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

$$\widetilde{f} = \frac{c_f}{f}, \qquad \qquad \widetilde{g} = \frac{c_g}{g}.$$
(32)

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

$$-\operatorname{div}(v(x)) = \tilde{f}(x) - \tilde{g}(x), \quad x \in \Omega, \quad \text{and} \quad v(x) \cdot \mathbf{n} = 0, \quad x \in \partial\Omega,$$
(33)

where 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

$$\frac{\partial \varphi(x,t)}{\partial t} = \eta(\varphi(x,t),t), \quad 0 \le t \le 1, \quad \varphi(x,0) = x,$$
(34)

with

$$\eta(y,s) := \frac{v(y)}{s\tilde{f}(y) + (1-s)\tilde{g}(y)}, \quad y \in \Omega, \ s \in [0,1].$$
(35)

4. Get the new grid points via

$$\varphi(x) := \varphi(x, 1). \tag{36}$$

In order to adapt the grid to an object, we have to choose the monitor function f(x) accordingly. In the formulation of the grid deformation algorithm the cell size distribution is the quantity of interest, because it allows us to control the grid deformation. We want cells of a small size near the object and we can allow bigger cells in regions located far from the object. Taking these observations into account, a reasonable choice for the monitor function is the distance of the nodes of the grid to the surface of the object, which is our choice for the following examples. In the case of surface triangulations the question arises how to calculate the distance function in an efficient way. Efficient algorithms for this task are Fast Marching or Fast Sweeping methods [24], additionally we developed an efficient Branch & Bound algorithm [25].

6.2 Numerical Examples

We apply the grid deformation in a stationary simulation of a flow around a torusknot, which is represented as surface triangulation. The computational grid has the following dimensions $\Omega = (0, 2.2) \times (0, 0.41) \times (0, 0.41)$, the number of nodes, elements and degrees of freedom at refinement level 4 are 75.625, 69.120 and 213.696. We compare results with grid deformation to those without grid deformation (see Fig. 7) in terms of quality of geometry approximation. Furthermore, we compare the results on level 4 with grid deformation to results on level 5 without grid deformation. As Fig. 9 shows, the geometry approximation is clearly better in the case of the adapted mesh. Even the level 5 result does not have the smoothness of the level 4 result with grid deformation. This visual observation is supported by the number of degrees of freedom that are used to model the inside of the object. Without grid deformation at level 4 only 1617 DOFs are located inside the object, with grid deformation 16109 DOFs are inside the object, at level 5 without grid deformation 13038 are inside.

6.3 Concluding Remarks

The results in this chapter show that the described grid deformation method provides a much better geometry approximation. For the presented case the grid deformation method

provides an increase in geometry approximation that is comparable to the increase that an additional level in grid refinement would yield. Our results in the stationary case suggest that we can expect to see the same beneficiary behaviour in instationary particulate flow simulations. The increase in geometry resolution in turn is expected to yield an accuracy improvement in the drag, lift and torque calculations. In our approach to the particulate flow problem these are the main quantities governing the movement of the particles. Hence, we expect to observe an overall improvement of our method. The topic of our following work will be to analyze the combination of the FBM-FEM and grid deformation in much more detail and more quantitatively than we have done in this article.



(a) Length section of the grid at level 4 with grid deformation



(b) Length section of the grid at level 4 without grid deformation

Figure 7: The figure shows the original surface triangulation, the adapted mesh and in contrast the non-adapted mesh



Figure 8: Geometry approximation and flow field at level 4 without grid deformation



Figure 9: Geometry approximation and flow field at level $4\ {\rm with}\ {\rm grid}\ {\rm deformation}$



Figure 10: Geometry approximation and flow field at level 5 (578641 nodes) without grid deformation

REFERENCES

- [1] Ardekani, A.M.; Rangel R.H.: Numerical investigation of particle-particle and particle-wall collisions in a viscous fluid, *Journal of Fluid Mechanics*, Volume 596, 437-466, 2008
- [2] Blasco, J.; Calzada,C.; Marin, M.: A Fictitious Domain, parallel numerical method for rigid particulate flows, *Journal of Computational Physics*, 228, 20, 2009, 7596-7613
- [3] S. Boenisch, V. Heuveline, On the numerical simulation of the instationary free fall of a solid in a fluid. I. The Newtonian case, Computers and Fluids 36(9), 2007, 1434-1445.
- [4] 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)
- [5] Clift,R., Grace, J.R., and Weber, M.E.: Bubbles, Drops, and Particles, Academic Press, New York, 1978
- [6] Coquerelle, M., Cottet, G.-H.: A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies, *Journal of Computational Physics*, 2008, doi:10.1016/j.jcp.2008.03.041
- [7] 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)
- [8] Dacorogna, B., Moser, J.: On a Partial Differential Equation Involving the Jacobian Determinant. Annales de le Institut Henri Poincare, 7, 1 – 26 (1990)
- [9] Fortes, A., Joseph, D.D., Lundgren, T.: Nonlinear mechanics of fluidization of beds of spherical particles, J. Fluid Mech., 177, 497 – 483 (1987)
- [10] Galdi, G.P., Heuveline, V.: Lift and Sedimentation of particles on the Flow of a Viscoelastic Liquid in a Channel Proc. Workshop "Free and Moving Boundaries: Analysis, Simulatiion and Control", Dec. 2-4, 2004, Houston TX, USA
- [11] 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)
- [12] 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)
- [13] 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)

- [14] 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)
- [15] Grajewski, M.; Köster, M.; Turek, S.: Mathematical and numerical analysis of a robust and efficient grid deformation method in the finite element context, SIAM Journal on Scientific Computing, 31, 2, 1539–1557, 2008
- [16] Grajewski, M.; Köster, M.; Turek, S.: Numerical analysis and implementational aspects of a new multilevel grid deformation method, Applied Numerical Mathematics, 60, 8, 767-781, doi:10.1016/j.apnum.2010.03.017, 2010
- [17] Hein, M.: Effiziente Distanzberechnung und Punktklassifizierung für NURBS-basierte CAD-Modelle zur Gitteradaption in der numerischen Strömungssimulation, DA, TU Dortmund, 2007
- [18] Hu, H.H., Joseph, D.D., Crochet, M.J.: Direct Simulation of Fluid Particle Motions. Theor. Comp. Fluid Dyn., 3, 285 – 306 (1992)
- [19] 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)
- [20] Lefebvre A.: Numerical simulation of gluey particles, ESAIM: Mathematical Modelling and Numerical Analysis, 43:53-80 (2009)
- [21] 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)
- [22] Maury, B.: Direct Simulations of 2D Fluid-Particle Flows in Biperiodic Domains. J. Comput. Phy., 156, 325 – 351 (1999)
- [23] Maury, B.: A time-stepping scheme for inelastic collisions, Numerische Mathematik, Volume 102, Number 4, pp. 649 - 679, 2006
- [24] Miemczyk, M.:Hexaeder-Gittergenerierung durch Kombination von Gitterdeformations-, Randadaptions- und Fictitious-Boundary-Techniken zur Strömungssimulation um komplexe dreidimensionale Objekte, DA, TU Dortmund, 2007
- [25] Münster, R.: Effiziente dynamische Suchdatenstrukturen zur Distanzberechnung in numerischen Strömungssimulationen, DA, TU Dortmund, 2007
- [26] 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)

- [27] T.-W. Pan, D.D. Joseph, R. Bai, R. Glowinski, V. Sarin: Fluidization of 1204 spheres: simulation and experiments, J. Fluid Mech. 451, 2002, 169-191
- [28] Patankar, N.A., Sharma, N: A fast projection scheme for the direct numerical simulation of rigid particulate flows, *Communications in Numerical Methods in Engineering*, 21, 2005, 1509-1524
- [29] Patankar, N.A., Sharma, N: A fast computation technique for the direct numerical simulation of rigid particulate flows, *Journal of Computational Physics*, 205, 2, 2005, 439 -457
- [30] 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)
- [31] Wan, D.; Turek, S.: Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method, International Journal for Numerical Methods in Fluids, 51, 5, 531–566, Published online 2005 in Wiley InterScience (www.interscience.wiley.com). doi: L 10.1002/fld.1129, 2006
- [32] Wan, D.C., Turek, S.: An Efficient Multigrid-FEM Method for the Simulation of Liquid-Solid Two Phase Flows. Journal of Computational and Applied Mathematics, 203, 2, 561– 580, 2007
- [33] 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)
- [34] Wan, D.; Turek, S.: An efficient multigrid-FEM method for the simulation of solid-liquid two phase flows, J. Comput. Appl. Math., 203, 2, 561-580, 2007.
- [35] Wan, D.; Turek, S.: Direct Numerical Simulation of Particulate Flow via Multigrid FEM Techniques and the Fictitious Boundary Method. *International Journal of Numerical Meth*ods in Fluids, 51, 5, 531-566, 2006
- [36] Wan, D.; Turek, S.: Fictitious Boundary and Moving Mesh Methods for the Numerical Simulation of Rigid Particulate Flows, J. Comput. Phys., 22, 28-56, 2006.
- [37] Turek, S.: Efficient Solvers for Incompressible Flow Problems. Springer Verlag, Berlin-Heidelberg-New York (1999)
- [38] 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)

- [39] Liao, G., Semper, B.: A Moving Grid Finite–Element Method Using Grid Deformation. Numerical Methods for Partial Differential Equations, **11**, 603 – 615 (1995)
- [40] 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)