Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 192 (2003) 105–123
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fictitious domain/finite element method for particulate flows

C. Diaz-Goano [a,*], P.D. Minev [b], K. Nandakumar [a]

[a] *Department of Chemical and Materials Engineering, 536 Chemical and Materials Engineering Building, University of Alberta, Edmonton, Alb., Canada T6G 2G6*
[b] *Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alb., Canada T6G 2G1*

Received 25 November 2002; received in revised form 5 May 2003; accepted 24 June 2003

Abstract

The paper presents a finite element method for the direct numerical simulation of 3D incompressible fluid flows with suspended rigid particles. It uses the fictitious domain approach extending the fluid domain into the domain occupied by the particles. Although the idea for the present formulation came from the fictitious domain formulation of Glowinski et al. [J. Comput. Phys. 169 (2001) 363] it finally yielded a new scheme which allows for a more efficient integration of the system of equations and which is easier to implement algorithmically. It also avoids the need to grid the rigid particles and solves the entire problem on a single Eulerian grid. The spatial discretization is performed with $\mathbb{P}_2 - P_1$ finite elements but any other stable pair of approximations can be used. The present scheme is compared to a second order scheme based on the approach in Glowinski et al. [loc. cit.] using the test case of a single sphere in a viscous liquid. Its accuracy/costs performance is excellent.

AMS: 65N35; 65N22; 65F05; 35J05

Keywords: Finite element method; Fictitious domain method; Particulate flow

1. Introduction

The mathematical description of multiphase flows, similarly to the mathematical description of turbulence, is still a significant challenge for the mathematicians and engineers working in this area. The models based on the interpenetrating continua hypothesis suffer of similar closure problems as the Reynolds-averaged turbulence models. Therefore, in the recent years some efforts were concentrated on the direct simulation of multiphase flows which would allow, again similarly to the turbulence case, for a more accurate solution of the closure problems. This is a very demanding computational task which requires the development of new discretization techniques and computer implementations.

* Corresponding author. Tel.: +780-987-8745; fax: +780-987-5349.
E-mail addresses: carolina@ualberta.ca (C. Diaz-Goano), minev@ualberta.ca (P.D. Minev), kumar.nandakumar@ualberta.ca (K. Nandakumar).

0021-9991/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00349-8

Generally speaking, there are two main classes of methods for direct simulations of multiphase flows. The first type are methods which discretize the Navier–Stokes equations on a fixed, Eulerian grid and move the particles in this grid. Probably the most prominent work in this direction (in the case of flows with distributed rigid particles) was done by a group of researchers from the Universities of Houston and Minnesota (see [1,2] and the references therein). In the case of distributed fluid particles, a similar approach was introduced by Tryggvason and collaborators (see [3]). The second type of methods uses moving grids and usually requires remeshing and re-interpolation. Examples of such techniques are the one developed by Hu [4] and Johnson and Tezduyar [5]. It is hard to judge and compare the merits of these two approaches. In our view, the first approach is more attractive because it allows for the use of structured grids which are much more preferable in view of the parallelization of the algorithm. Moreover, it completely avoids the need of a mesh generation in very complex domains which is also a challenging algorithmic problem. Therefore, in a preliminary study we used the fictitious domain approach of [2] to develop a formally second order (in space and time) finite element scheme (see [6]). Then we modified it, in an attempt to make it more efficient and as a result we devised a new scheme, still based on the fictitious domains idea. The numerical experiments showed a very satisfactory performance of this scheme. Its key novelty is that it approximates both, the fluid velocity field and the Lagrange multiplier for imposition of the rigid body motion on the same fixed Eulerian grid. This allowed us to avoid the gridding of the rigid particles. Moreover, using a properly weighted $\mathbb{H}^1$ norm for the imposition of the rigid body motion, we were able to reduce the work for computing the fluid/rigid particles velocities and enhance the stability of the algorithm (this was verified only experimentally). The derivation of this scheme and its assessment is a subject of the rest of this paper.

In Section 2, we discuss the fictitious domain approach and the Distributed Lagrange Multipliers (DLM) method of Glowinski et al. [2], and derive the starting formulation for the present scheme. There we also describe the splitting algorithm and the finite element discretization with $\mathbb{P}_2 - P_1$ elements. In Section 3, we compare the results of the simulation of a rigid sphere falling in a viscous liquid with available experimental data and results obtained with the scheme described in [6]. In addition, two other simulations are presented there: a spherical particle in a rotating cylinder and two spherical particles settling under gravity and interacting with each other (drafting, kissing and tumbling). Finally, the conclusions are summarized in Section 4.

2. A fictitious domains formulation using a global Lagrange multiplier

2.1. Governing equations

We consider a bounded domain Ω_1 with an external boundary Γ filled with a Newtonian liquid of density ρ_1 and viscosity μ_1. Within this liquid we consider n rigid particles occupying a domain $\Omega_2 = \bigcup_{i=1}^{n} \Omega_{2,i}$ and having densities $\rho_{2,i},\ i = 1, \ldots, n$. Let us also denote the interface between Ω_1 and Ω_2 by Σ and the entire domain filled with the fluid and the particles by $\Omega = \Omega_1 \cup \Omega_2$. The equations of motion of the fluid in Ω_1 are the Navier–Stokes equations (presented here in a dimensionless, stress-divergence form)

$$\rho_1 \frac{\mathrm{D}\hat{\mathbf{u}}_1}{\mathrm{D}t} = \nabla \cdot \hat{\boldsymbol{\sigma}}_1, \quad \nabla \cdot \hat{\mathbf{u}}_1 = 0 \quad \text{in } \Omega_1, \tag{1}$$

where $\hat{\mathbf{u}}_1$ is the velocity, $\mathrm{D}/\mathrm{D}t$ denotes the full derivative in time (including the advection terms), $\hat{\boldsymbol{\sigma}}_1$ is the stress tensor and ρ_1 is the fluid density. $\hat{\boldsymbol{\sigma}}_1$ is defined as usual as $\hat{\boldsymbol{\sigma}}_1 = \hat{p}_1 \boldsymbol{\delta} + 2\mu_1 \mathbb{D}[\hat{\mathbf{u}}_1]$ with $\hat{p}_1$ being the pressure in the liquid phase, $\mathbb{D}[\hat{\mathbf{u}}_1] = 0.5[\nabla \hat{\mathbf{u}}_1 + (\nabla \hat{\mathbf{u}}_1)^{\mathrm{T}}]$ being the rate-of-strain tensor and $\boldsymbol{\delta}$ being the Kronecker tensor. The boundary conditions on Γ are not of a concern for the present method and therefore we assume the simplest case of homogeneous Dirichlet conditions. On the internal boundary of Ω_1, Σ, we

presume a no-slip condition for the velocity. The problem requires an initial condition for the velocity which we presume to be in the form $\hat{\mathbf{u}}_1(\mathbf{x},0)=\mathbf{u}_0$, $\nabla\cdot\mathbf{u}_0=0$. The equations of motion of a rigid particle are usually written in terms of the velocity of its centroid $\mathbf{U}_i$ and its angular velocity $\boldsymbol{\omega}_i$, $i=1,\ldots,n$. They read

$$M_i\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t}=M_i\mathbf{g}+\mathbf{F}_i, \tag{2}$$

$$\mathbf{I}_i\frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t}+\boldsymbol{\omega}_i\times\mathbf{I}_i\boldsymbol{\omega}_i=\mathbf{T}_i, \tag{3}$$

where M_i is the particle mass, $\mathbf{g}$ is the gravity acceleration, $\mathbf{F}_i$ is the total hydrodynamic force acting on it, $\mathbf{I}_i$ is its tensor of inertia, and $\mathbf{T}_i$ is the hydrodynamic torque about its center of mass.

2.2. Fictitious domain approach and the DLM method

Before the presentation of the new technique, we would like to briefly review the fictitious domain approach and its implementation in the DLM method of Glowinski et al. [2]. The fictitious domain approach extends the fluid equations of motion into the domain occupied by the particles, Ω_2, and imposes the rigid body motion described by Eqs. (2) and (3) onto the restriction of the so extended velocity field. The DLM method enforces this constraint by introducing localized (distributed within each particle) Lagrange multipliers $\boldsymbol{\zeta}_i$, and relaxing the constraint by means of a weak formulation. After an extensive manipulation of the governing equations of both, the liquid and the particle phases, the following combined variational principle is derived in [2]: [1] *Find* $\mathbf{u}\in\mathbb{H}^1_0(\Omega), p\in L^2_0(\Omega)=\{q\in L^2(\Omega)\mid\int_\Omega q\,\mathrm{d}\mathbf{x}=0\},\boldsymbol{\zeta}\in\mathbb{H}^1(\Omega_2),\mathbf{U}\in R^3$, and $\boldsymbol{\omega}\in R^3$ *that satisfy:*

$$\begin{aligned}&\int_\Omega\rho_1\left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}-\mathbf{g}\right)\cdot\mathbf{v}\,\mathrm{d}\mathbf{x}+(1-\rho_1/\rho_2)\left(M\left(\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t}-\mathbf{g}\right)\cdot\mathbf{V}+\mathbf{I}\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}\cdot\boldsymbol{\xi}\right)+\int_\Omega\boldsymbol{\sigma}:\mathbb{D}[\mathbf{v}]\,\mathrm{d}\mathbf{x}\\&\quad=\langle\boldsymbol{\zeta},\mathbf{v}-(\mathbf{V}+\boldsymbol{\xi}\times\mathbf{r})\rangle_{\Omega_2}\quad\forall\mathbf{v}\in\mathbb{H}^1_0(\Omega),\ \ \mathbf{V}\in R^3,\ \ \boldsymbol{\xi}\in R^3,\end{aligned} \tag{4}$$

$$\int_\Omega q\nabla\cdot\mathbf{u}\,\mathrm{d}\mathbf{x}=0\quad\forall q\in L^2(\Omega), \tag{5}$$

$$\langle\eta,\mathbf{u}-(\mathbf{U}+\boldsymbol{\omega}\times\mathbf{r})\rangle_{\Omega_2}=0\quad\forall\eta\in\mathbb{H}^1(\Omega_2), \tag{6}$$

$$\mathbf{u}\,|_{t=0}=\mathbf{u}_0,\quad\mathbf{U}\,|_{t=0}=\mathbf{U}_0,\quad\boldsymbol{\omega}\,|_{t=0}=\boldsymbol{\omega}_0. \tag{7}$$

Here $\mathbf{v}$, $\mathbf{V}$ and $\boldsymbol{\xi}$ are properly chosen test functions for the fluid velocity, particle velocity and the particle angular velocity equations, $\mathbf{r}=\mathbf{x}-\mathbf{X}$ with $\mathbf{X}$ being the coordinate vector of the centroid of the particle, and $\boldsymbol{\sigma}$ is a properly extended to Ω_2 stress tensor. The fluid velocity is discretized on a fixed (Eulerian) grid that covers the entire domain Ω and the Lagrange multiplier $\boldsymbol{\zeta}$ is discretized on a separate grid that covers the particle. The combined equations (4)–(7) are discretized in time and split into the following substeps:

[1] In [2], the combined equation is derived for the case of one particle only ($M_i=M$, $\rho_{2,i}=\rho_2$, $\mathbf{U}_i=\mathbf{U}$, $\boldsymbol{\omega}=\boldsymbol{\omega}_i$, $\boldsymbol{\zeta}=\boldsymbol{\zeta}_i$), but its generalization for the multiparticle case is straightforward.

2.2.1. Incompressibility constraint

At this substep, a fluid velocity field and pressure, based on the Eulerian grid, are computed so that the velocity is divergence free (see [2, Eqs. (45) and (46)]). This is in fact a Darcy problem which is resolved in [2] by means of an Uzawa/conjugate-gradient preconditioned iteration.

2.2.2. Advection–diffusion

At this substep, the incompressible fluid velocity is further advected and diffused (see [2, Eq. (47)]). Further, the particle velocity and the position of its centroid are predicted [2, Eqs. (48) and (49)].

The first two substeps constitute a splitting algorithm for the Navier–Stokes equations which is very similar to the so-called velocity-correction projection. The only difference is that in this algorithm, the incompressibility constraint is strictly enforced via Uzawa iteration while the velocity-correction projection method does only one step of this iteration using the same preconditioner (a homogeneous Neumann boundary value problem for the pressure Poisson equation). Since it has been proved that the velocity-correction method achieves optimal error in time for the velocity (see [7]) there is no need for additional iterations in the first substep since in the second substep the incompressibility of the fluid velocity is destroyed anyway. It is further influenced in the next substep which enforces the rigid body motion. Therefore, in [6], in the first two substeps we used a second order pressure-correction projection method combined with the method of characteristics.

2.2.3. Rigid body constraint

This is the substep that enforces the side constraint, Eq. (6). It is done in a typical for such constraints inner/outer iteration loop. The correction to the fluid velocity $\mathbf{u}$ is computed in the outer loop and then used to compute the correction to the Lagrange multipliers $\boldsymbol{\zeta}$ in the inner loop (see [2, Eqs. (62)–(74)]). The DLM method uses a conjugate gradient iteration for both, the inner and the outer loop. Note that in this inner/outer loop, one needs to solve one linear system for the fluid velocity and one linear system for the restriction of $\boldsymbol{\zeta}$ over each of the particle domains $\Omega_{2,i}$ at each iteration step. It is precisely at this step where the present technique differs most significantly from the DLM method. By using a globally rather than a locally defined Lagrange multiplier and enforcing the constraint given by Eq. (6) with a properly weighted inner product, it avoids the need to solve the linear systems over each of the particles. Instead, it solves just one linear system for the globally defined Lagrange multiplier, whose size is equal to the size of the fluid velocity system solved in the outer iteration of DLM. The savings in CPU time of the present procedure clearly grow linearly with the number of particles. The memory for the storage of one extra global vector is usually less than the memory for the storage of the additional connectivity tables and matrices for the Lagrange multipliers of the DLM method (even for the case of one particle). Note that the matrices of the problem for the global Lagrange multiplier in the present procedure are the same as the matrices of the problem that defines the fluid velocity.

The new fictitious domain formulation and its numerical discretization are presented in the next two subsections.

2.3. A new fictitious domain formulation

For the derivation of the new formulation, it is more convenient first to define a velocity field $\mathbf{u}_2(\mathbf{x},t)$ which is continuous within the region occupied by any given particle, and zero in Ω_1. Clearly, $\mathbf{u}_2 \in \mathbb{L}^2(\Omega)$, and $\mathbf{u}_2(\mathbf{x},t)|_{\Omega_i} = \mathbf{U}_i(t) + \boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t))$. The momentum equation for the restriction of $\mathbf{u}_2$ on $\Omega_{2,i}$, $\mathbf{u}_{2,i}$, can be written in the following integral form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{2,i}} \rho_{2,i}\mathbf{u}_2 \,\mathrm{d}\Omega = \int_{\Omega_{2,i}} (\rho_{2,i} - \rho_1)\mathbf{g} \,\mathrm{d}\Omega + \int_{\partial\Omega_{2,i}} \hat{\boldsymbol{\sigma}}_1 \mathbf{n}_i \,\mathrm{d}s \tag{8}$$

with $\mathbf{n}_i$ being the outward normal to the surface of the ith particle. The last term in this equation represents the total hydrodynamic force acting on the surface of the ith particle. Let us denote by $\boldsymbol{\sigma}_1$ the continuous extension of the stress $\hat{\boldsymbol{\sigma}}_1$ over the entire domain Ω. Such an extension can always be constructed and this is the basis for the different fictitious domain methods that are proposed in the literature. Then using the divergence theorem we can rewrite Eq. (8) in the following form:

$$\int_{\Omega_{2,i}} \frac{\mathrm{D}}{\mathrm{D}t}(\rho_{2,i}\mathbf{u}_2)\,\mathrm{d}\Omega = \int_{\Omega_{2,i}} (\rho_{2,i}-\rho_1)\mathbf{g}\,\mathrm{d}\Omega + \int_{\Omega_{2,i}} \nabla\cdot\boldsymbol{\sigma}_1\,\mathrm{d}\Omega. \tag{9}$$

A natural way to extend the stress tensor in $\Omega_{2,i}$ is to assume that it is Newtonian and extend the velocity field $\hat{\mathbf{u}}_1$ to $\mathbf{u}_1$ such that $\mathbf{u}_1|_{\Omega_1} = \hat{\mathbf{u}}_1$, the pressure to some p_1 such that $p_1|_{\Omega_1} = \hat{p}_1$, and write $\nabla\cdot\boldsymbol{\sigma}_1 = -\nabla p_1 + \mu_1\nabla^2\mathbf{u}_1$. Then, Eq. (9) can be rewritten as

$$\int_{\Omega_{2,i}} \frac{\mathrm{D}}{\mathrm{D}t}(\rho_{2,i}\mathbf{u}_{2,i})\,\mathrm{d}\Omega = \int_{\Omega_{2,i}} (\rho_{2,i}-\rho_1)\mathbf{g}\,\mathrm{d}\Omega + \int_{\Omega_{2,i}} \left(-\nabla p_1 + \mu_1\nabla^2\mathbf{u}_1\right)\mathrm{d}\Omega. \tag{10}$$

Now if we adopt the notation

$$\mathbf{F} = \begin{cases} -\rho_1 \frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t} + \mu_1\nabla^2\mathbf{u}_1 - \nabla p_1 & \text{in } \Omega_{2,i},\ i=1,\ldots,n, \\ 0 & \text{in } \Omega_1 \end{cases} \tag{11}$$

the equation for the ith particle momentum becomes

$$\int_{\Omega_{2,i}} \frac{\mathrm{D}}{\mathrm{D}t}(\rho_{2,i}\mathbf{u}_2 - \rho_1\mathbf{u}_1)\,\mathrm{d}\Omega = \int_{\Omega_{2,i}} [(\rho_{2,i}-\rho_1)\mathbf{g} + \mathbf{F}]\,\mathrm{d}\Omega. \tag{12}$$

Because of Eq. (11), we can now extend the momentum equation (1) to the entire Ω as

$$\rho_1 \frac{D\mathbf{u}_1}{Dt} = -\nabla p_1 + \mu_1\nabla^2\mathbf{u}_1 - \mathbf{F}, \quad \nabla\cdot\mathbf{u}_1 = 0 \quad \text{in } \Omega. \tag{13}$$

The additional force per unit volume $\mathbf{F}$ in this equation can be interpreted as the interaction force between the two phases. As discussed by Glowinski et al. [2] it enforces the rigid body motion onto the fluid velocity field within each of the particles. If we impose this additional constraint onto $\mathbf{u}_1$, i.e.,

$$\mathbf{u}_1 = \mathbf{u}_2 \quad \text{in } \Omega_{2,i} \tag{14}$$

then momentum equation for the ith particle, Eq. (12), can be written as

$$\int_{\Omega_{2,i}} \frac{\mathrm{D}}{\mathrm{D}t}[(\rho_{2,i}-\rho_1)\mathbf{u}_2]\,\mathrm{d}\Omega = \int_{\Omega_{2,i}} [(\rho_{2,i}-\rho_1)\mathbf{g} + \mathbf{F}]\,\mathrm{d}\Omega. \tag{15}$$

Now we recall that $\mathbf{u}_2$ is a rigid body velocity field, i.e., $\mathbf{u}_2(\mathbf{x},t)|_{\Omega_i} = \mathbf{U}_i(t) + \boldsymbol{\omega}_i(t)\times(\mathbf{x}-\mathbf{X}_i(t))$. Then Eq. (15) implies the following equation for $\mathbf{U}_i$

$$\Delta M_i \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \Delta M_i\mathbf{g} + \int_{\Omega_{2,i}} \mathbf{F}\,\mathrm{d}\Omega, \tag{16}$$

where $\Delta M_i = \int_{\Omega_{2,i}}(\rho_{2,i}-\rho_1)\,\mathrm{d}\Omega$. The angular velocity $\boldsymbol{\omega}_i$ can be recovered from the no-slip boundary condition on the surface of the ith particle. It reads

$$\mathbf{U}_i(t) + \boldsymbol{\omega}_i(t)\times(\mathbf{x}-\mathbf{X}_i(t)) = \mathbf{u}_1 \quad \text{on } \partial\Omega_{2,i}.$$

Then clearly

$$\int_{\partial\Omega_{2,i}} (\boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t)) \times \mathbf{n}\,\mathrm{d}s = \int_{\partial\Omega_{2,i}} (\mathbf{u}_1 - \mathbf{U}_i(t)) \times \mathbf{n}\,\mathrm{d}s$$

which yields, using the Stokes theorem, that

$$\boldsymbol{\omega}_i(t) V_{\Omega_i} = 0.5 \int_{\Omega_{2,i}} \nabla \times (\mathbf{u}_1 - \mathbf{U}_i(t))\,\mathrm{d}\Omega,$$

where V_{Ω_i} is the volume of the ith particle. Finally, the strong formulation of the present version of the fictitious domain method is given by

$$\rho_1 \frac{D\mathbf{u}_1}{Dt} = -\nabla p_1 + \mu_1 \nabla^2 \mathbf{u}_1 - \mathbf{F}, \quad \nabla \cdot \mathbf{u}_1 = 0 \quad \text{in } \Omega, \tag{17}$$

$$\Delta M_i \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \Delta M_i \mathbf{g} + \int_{\Omega_{2,i}} \mathbf{F}\,\mathrm{d}\Omega, \quad i = 1, \ldots, n, \tag{18}$$

$$\boldsymbol{\omega}_i(t) V_{\Omega_i} = 0.5 \int_{\Omega_{2,i}} \nabla \times (\mathbf{u}_1 - \mathbf{U}_i(t))\,\mathrm{d}\Omega, \quad i = 1, \ldots, n, \tag{19}$$

$$\mathbf{U}_i(t) + \boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t)) = \mathbf{u}_1 \quad \text{in } \Omega_{2,i}, \quad i = 1, \ldots, n. \tag{20}$$

From this set of equations, it is clear that $\mathbf{F}$ is a term that enforces the constraint given by Eq. (20) onto $\mathbf{u}_1$ and $\mathbf{u}_2$. It is non-zero only within the domain occupied by the particles and therefore it is closely related to the distributed Lagrange multiplier of Glowinski et al. [2]. However, its impact on the fluid velocity field is global, over the entire domain, due to Eq. (17). This led us to the idea to define another, global Lagrange multiplier, $\boldsymbol{\lambda}$, which is related to $\mathbf{F}$ through the following boundary value problem

$$\begin{aligned} &-\alpha\boldsymbol{\lambda} + \mu_1 \nabla^2 \boldsymbol{\lambda} = \mathbf{F} \quad \text{in } \Omega, \\ &\boldsymbol{\lambda} = 0 \quad \text{on } \Gamma, \end{aligned} \tag{21}$$

where $\alpha > 0$ is a constant to be defined later. This is a well-posed problem for $\mathbf{F} \in \mathbb{L}^2$, and it is more convenient to use its unique solution to impose the constraint of Eq. (20). The benefit of using it, is that it has the same regularity as $\mathbf{u}_1$ and therefore they can be discretized on the same grid (i.e., discretization space). With this definition of $\boldsymbol{\lambda}$ and a proper choice for α, as it will become clear in the following section, we can significantly decrease the computational expenses for the iterative solution of the system of equations (17)–(20).

Substituting $\mathbf{F}$ from Eq. (21) into Eqs. (17)–(20) we obtain the following system which will be used further to compute $\mathbf{u}_1$, $\mathbf{U}_i$ and $\boldsymbol{\omega}_i$

$$\rho_1 \frac{D\mathbf{u}_1}{Dt} = -\nabla p_1 + \mu_1 \nabla^2 \mathbf{u}_1 + \alpha\boldsymbol{\lambda} - \mu_1 \nabla^2 \boldsymbol{\lambda}, \quad \nabla \cdot \mathbf{u}_1 = 0 \quad \text{in } \Omega, \tag{22}$$

$$\Delta M_i \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \Delta M_i \mathbf{g} - \int_{\Omega_{2,i}} \alpha\boldsymbol{\lambda}\,\mathrm{d}\Omega - \mu_1 \int_{\partial\Omega_{2,i}} \frac{\partial\boldsymbol{\lambda}}{\partial\mathbf{n}}\,\mathrm{d}s, \quad i = 1, \ldots, n, \tag{23}$$

$$V_{\Omega_i} \boldsymbol{\omega}_i = 0.5 \int_{\Omega_{2,i}} \nabla \times (\mathbf{u}_1 - \mathbf{U}_i)\,\mathrm{d}\Omega, \quad i = 1, \ldots, n, \tag{24}$$

$$\mathbf{U}_i(t) + \boldsymbol{\omega}_i(t) \times (\mathbf{x} - \mathbf{X}_i(t)) = \mathbf{u}_1 \quad \text{in } \Omega_{2,i}, \quad i = 1, \ldots, n. \tag{25}$$

Note that in the second equation we have integrated $\nabla^2 \boldsymbol{\lambda}$ by parts. Note also that this choice for $\boldsymbol{\lambda}$ is similar to the choice of Lagrange multiplier in [2] if the inner product for imposition of the rigid body constraint is chosen to be the $\mathbb{H}^1$ inner product. In the present case, however, $\boldsymbol{\lambda}$ is defined over the entire computational domain while in [2] it is defined only locally, within the areas occupied by the particles. The global Lagrange multiplier will allow for a change in the fictitious domain method as proposed in [1,2], which generally increases the efficiency of the method and simplifies it (see the following section).

In addition to the system of equations (22)–(25) we also need to solve an equation for the position of the center of mass of each particle which reads

$$\frac{\partial \mathbf{X}_i}{\partial t} = \mathbf{U}_i, \quad i = 1, \ldots, n.$$

Remark 1. An idea for using the fluid velocity grid points inside the particles together with surface control points for imposition of the rigid body motion is given in [1, Remark 6.4]. The suggestion there is to use Dirac δ-functions to span the Lagrange multipliers inside the particles.

Remark 2. Remark 7.3 of [1] suggests to compute the particles centroidal and angular velocities in a way similar to the one suggested above.

2.4. Discretization procedure

We discretize the system of equations (22)–(25) using a finite element procedure although finite difference or finite volume based procedures can also be used. Note that only Eqs. (22) and (25) require weak formulations. Eqs. (23) and (24) are an ordinary differential and an algebraic equation. The overall problem is in fact a Navier–Stokes problem with an additional constraint given by Eq. (25), and therefore we chose to discretize it with a projection scheme (for $\mathbf{u}_1$ and p_1) combined with an additional iteration for the imposition of the rigid body constraint. The scheme is derived from the (formally) second order in time characteristic/projection scheme described in [8]. In space we used $\mathbb{P}_2 - P_1$ finite elements which resulted in a formally second order spatial discretization for the velocity. The algorithm can be summarized in the following four steps.

2.4.1. Substep 1 (advection)

The advective part of the system is integrated with the method of characteristics [8]. If $\underline{\mathbf{x}}$ is an approximation of the foot of the characteristic originating at $\mathbf{x}$ then the advected velocity is given by $\tilde{\mathbf{u}}_1^n(\mathbf{x}) = \mathbf{u}_1^n(\underline{\mathbf{x}}), \tilde{\mathbf{u}}_1^{n-1}(\mathbf{x}) = \mathbf{u}_1^{n-1}(\underline{\mathbf{x}})$. $\underline{\mathbf{x}}$ is usually approximated with an Euler explicit scheme [8].

The center of mass of the ith particle is predicted explicitly as

$$\mathbf{X}_i^{p,n+1} = \mathbf{X}_i^{n-1} + 2\delta t \mathbf{U}_i^n, \tag{26}$$

where δt is the time step.

2.4.2. Substep 2 (diffusion)

If we set $\tau_0 = 3/(2\delta t)$, $\tau_1 = -2/\delta t$, $\tau_2 = 1/(2\delta t)$, then this substep can be written as

$$\begin{aligned} &\rho_1 \tau_0 \mathbf{u}_1^* - \mu_1 \nabla^2 \mathbf{u}_1^* = -\rho_1(\tau_1 \tilde{\mathbf{u}}_1^n - \tau_2 \tilde{\mathbf{u}}_1^{n-1}) - \nabla p^n \quad \text{in } \Omega, \\ &\mathbf{u}_1^* = 0 \quad \text{on } \Gamma. \end{aligned} \tag{27}$$

2.4.3. Substep 3 (incompressibility)

$$\begin{aligned}
&\tau_0\left(\mathbf{u}_1^{**}-\mathbf{u}_1^{*}\right)=-\nabla\left(p_1^{n+1}-p_1^{n}\right) \quad \text{in } \Omega,\\
&\nabla\cdot\mathbf{u}_1^{**}=0 \quad \text{in } \Omega,\\
&\mathbf{u}_1^{**}\cdot\mathbf{n}=0 \quad \text{on } \Gamma,
\end{aligned} \tag{28}$$

where $\mathbf{n}$ being the outward normal to Γ.

2.4.4. Substep 4 (rigid body constraint)

The rigid body motion is imposed iteratively using the following iteration. Let us first set the 0th approximation for $\boldsymbol{\lambda}^{n+1}$, $\mathbf{u}_1^{n+1}$, $\mathbf{U}_i^{n+1}$, $\boldsymbol{\omega}_i^{n+1}$ and $\mathbf{u}_2^{n+1}$ as

$$\begin{aligned}
&\boldsymbol{\lambda}^{0,n+1}=0,\\
&\mathbf{u}_1^{0,n+1}=\mathbf{u}_1^{**},\\
&\tau_0\mathbf{U}_i^{0,n+1}=-\tau_1\mathbf{U}_i^{n}-\tau_2\mathbf{U}_i^{n-1}+\mathbf{g},\\
&V_{\Omega_i}\boldsymbol{\omega}_i^{0,n+1}=0.5\int_{\Omega_i}\nabla\times\mathbf{u}_1^{0,n+1}\,\mathrm{d}\Omega,\\
&\mathbf{u}_2^{0,n+1}=\mathbf{U}_i^{0,n+1}+\boldsymbol{\omega}^{0,n+1}\times\left(\mathbf{x}-\mathbf{X}_i^{p,n+1}\right).
\end{aligned}$$

We also need to set $\boldsymbol{\lambda}^{1,n+1}=\boldsymbol{\lambda}^{n}$. If we denote the difference between two subsequent iterations for a quantity Q by δQ, i.e., $\delta Q^{k+1}=Q^{k+1}-Q^{k}$ then the subsequent iterates are computed for $k\geqslant 0$ by

$$\begin{cases}
(\rho_1\tau_0 I-\mu_1\nabla^2)\delta\mathbf{u}_1^{k+1,n+1}=(\alpha I-\mu_1\nabla^2)\delta\boldsymbol{\lambda}^{k+1,n+1} \quad \text{in } \Omega,\\
\delta\mathbf{u}_1^{k+1,n+1}=0 \quad \text{on } \Gamma,
\end{cases} \tag{29}$$

$$\begin{cases}
\Delta M_i\tau_0\delta\mathbf{U}_i^{k+1,n+1}=-\int_{\Omega_{2,i}}\alpha\delta\boldsymbol{\lambda}^{k+1,n+1}\,\mathrm{d}\Omega+\mu_1\int_{\partial\Omega_{2,i}}\frac{\partial\delta\boldsymbol{\lambda}^{k+1,n+1}}{\partial\mathbf{n}}\,\mathrm{d}s, \quad i=1,\ldots,n,\\
V_{\Omega_i}\boldsymbol{\omega}_i^{k+1,n+1}=0.5\int_{\Omega_i}\nabla\times\mathbf{u}_1^{k+1,n+1}\,\mathrm{d}\Omega, \quad i=1,\ldots,n,\\
\mathbf{u}_2^{k+1,n+1}=\mathbf{U}_i^{k+1,n+1}+\boldsymbol{\omega}_i^{k+1,n+1}\times\left(\mathbf{x}-\mathbf{X}_i^{p,n+1}\right) \quad \text{in } \Omega_{2,i}, \quad i=1,\ldots,n,\\
\mathbf{u}_2^{k+1,n+1}=0 \quad \text{in } \Omega_1
\end{cases} \tag{30}$$

$$\begin{cases}
\left(1+\frac{\rho_1}{\rho_{2,i}-\rho_1}\right)(\alpha I-\mu_1\nabla^2)\delta\boldsymbol{\lambda}^{k+2,n+1}=-(\rho_1\tau_0 I-\mu_1\nabla^2)\left(\mathbf{u}_1^{k+1,n+1}-\mathbf{u}_2^{k+1,n+1}\right) \quad \text{in } \Omega_{2,i}, \ i=1,\ldots,n,\\
\left(1+\frac{\rho_1}{\rho_{2,i}-\rho_1}\right)(\alpha I-\mu_1\nabla^2)\delta\boldsymbol{\lambda}^{k+2,n+1}=0 \quad \text{in } \Omega_1,\\
\delta\boldsymbol{\lambda}^{k+2,n+1}=0 \quad \text{on } \Gamma,
\end{cases} \tag{31}$$

where I is the identity operator. Note that in the last set of equations which determines the increment of $\boldsymbol{\lambda}$, we used the fact that $\mathbf{u}_2$ is a rigid body velocity field inside each particle and therefore $\nabla^2\mathbf{u}_2=0$ in $\Omega_{2,i}$.

Upon convergence for some $k=N$ we set $\mathbf{u}_1^{n+1}=\mathbf{u}_1^{N+1,n+1}$, $\mathbf{U}_i^{n+1}=\mathbf{U}_i^{N+1,n+1}$, $\boldsymbol{\omega}^{n+1}=\boldsymbol{\omega}^{N+1,n+1}$, $\boldsymbol{\lambda}^{n+1}=\boldsymbol{\lambda}^{N+1,n+1}$. The equation of the center of mass of the particles is solved with a second order predictor–corrector procedure, the predictor being given by Eq. (26), and the corrector being given by

$$\mathbf{X}_i^{n+1}=\mathbf{X}^{n}+0.5\delta t\left(\mathbf{U}_i^{n+1}+\mathbf{U}_i^{n}\right). \tag{32}$$

Similarly to the iteration in [2], instead of the present (Richardson) iteration we can also adopt a conjugate gradient type of iterative algorithm for imposition of the rigid body constraint. The numerical tests show that only few iterations per time step of the present procedure are necessary to obtain very reasonable results.

Note that in the above splitting algorithm only Eqs. (27)–(29) and (31) constitute boundary value problems for PDEs and in order to discretize them with finite elements we need to derive proper weak formulations. As it is usual for finite element projection schemes (see [9]) we choose $\mathbf{u}_1^*$ to be in $\mathbb{H}_0^1(\Omega)$ and p_1 to be in $\mathbb{H}^1(\Omega)$. The final velocity of *Substep 3*, $\mathbf{u}_1^{**}$, is also projected onto $\mathbb{H}_0^1(\Omega)$ as discussed in [9]. Since the Lagrange multiplier $\boldsymbol{\lambda}$ is a solution to a boundary value problem similar to the problem for $\mathbf{u}_1$, it is naturally chosen to be in $\mathbb{H}_0^1(\Omega)$. Actually it plays the role of a correction to both, the fluid and particle velocities. Of course, this correction will increase the divergence of the fluid field and in order to control better the divergence of this correction one may decompose the interaction force $\mathbf{F}$ into a divergence free part, $\boldsymbol{\lambda}$, and a curl free part, ∇p_2, and define both to be a solution of a Stokes-like problem. This would require to do one projection step for the additional pressure p_2 at each iteration for imposition of the rigid body motion. The present algorithm seems to be reasonably stable as it is, and therefore we did not try this more expensive alternative. Note that the DLM method has the same problem with the enforcement of the incompressibility constraint since the incompressible velocity field produced at its first substep ($\mathbf{u}^{n+1/3}$ in the notations of [2]) loses its incompressibility in the subsequent substeps of the spitting. The final field of this method ($\mathbf{u}^{n+1}$ in the notations of [2]) is not incompressible but the incompressibility can also be enforced in the way suggested above. In fact, the linear system for the fluid velocity that arises from the fictitious domain approach can be written in the form

$$\begin{bmatrix} A & L^{\mathrm{T}} & \Lambda^{\mathrm{T}} \\ L & 0 & 0 \\ \Lambda & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \\ \boldsymbol{\lambda}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \\ 0 \end{bmatrix},$$

where A is the total velocity matrix (a linear combination of the mass and stiffness matrices), L is the divergence matrix and Λ is the coupling matrix between the fluid velocity and the Lagrange multiplier $\boldsymbol{\lambda}$ (in the present method it reduces to the velocity mass matrix). This problem can be considered as an extended Stokes problem, and it can be resolved with a preconditioned inner/outer iteration similar to the one used to solve the Stokes problem alone (Uzawa iteration). The preconditioner for the pressure equations can be chosen to be a Laplace operator with homogeneous Neumann boundary conditions. The preconditioner for the equations for $\boldsymbol{\lambda}^{n+1}$ is given by Eq. (31). As a result of such an iteration both, the incompressibility and the rigid body constraints will be imposed very strictly but it will require the solution of one pressure Poisson problem and one problem of type of Eq. (31) in the inner step, and one problem involving the total matrix A in the outer step. This is an expensive procedure and the numerical results indicated that the present splitting/iterative procedure produces very accurate results and it is reasonably stable. If the stability becomes a problem, it is recommendable that the pressure p^{n+1} is corrected once more after *Substep 4*, repeating *Substep 3*.

The weak formulations of Eqs. (27) and (28) are given by

Find $\mathbf{u}_1^* \in \mathbb{H}_0^1(\Omega)$ *such that*

$$\int_\Omega \left(\rho_1 \tau_0 \mathbf{u}_1^* \mathbf{v} + \mu_1 \nabla \mathbf{u}_1^* \nabla \mathbf{v}\right) \mathrm{d}\Omega = -\int_\Omega \rho_1 (\tau_1 \tilde{\mathbf{u}}_1^n . + \tau_2 \tilde{\mathbf{u}}_1^{n-1}) \mathbf{v} \, \mathrm{d}\Omega + \int_\Omega p_1^n \nabla \cdot \mathbf{v} \, \mathrm{d}\Omega \quad \forall \mathbf{v} \in \mathbb{H}_0^1(\Omega). \tag{33}$$

Find $p_1^{n+1} \in \mathbb{H}^1(\Omega)$ *such that*

$$\int_\Omega \nabla \left(p_1^{n+1} - p_1^n\right) \nabla q \, \mathrm{d}\Omega = \tau_0 \int_\Omega \rho_1 \mathbf{u}_1^* \nabla q \, \mathrm{d}\Omega \quad \forall q \in \mathbb{H}^1(\Omega). \tag{34}$$

Find $\mathbf{u}_1^{**} \in \mathbb{H}_0^1(\Omega)$ *such that*

$$\int_\Omega \rho_1 \tau_0 \left(\mathbf{u}_1^{**} - \mathbf{u}_1^{*}\right)\mathbf{v}\,\mathrm{d}\Omega = \int_\Omega p_1^{n+1} \nabla \cdot \mathbf{v}\,\mathrm{d}\Omega \quad \forall \mathbf{v} \in \mathbb{H}_0^1(\Omega). \tag{35}$$

As discussed in [9] the last step can be eliminated but it usually reduces the divergence of the so computed velocity field. In the present context, since we do not control the divergence of the iterative correction $\boldsymbol{\lambda}$, it is advisable to solve Eq. (35). Before we give the weak formulation of Eq. (31) we note that if we choose $\alpha = \rho_1 \tau_0$ the problem described by Eq. (29) can easily be resolved and the solution is

$$\mathbf{u}_1^{k+1,n+1} = \mathbf{u}_1^{k,n+1} + (\boldsymbol{\lambda}^{k+1,n+1} - \boldsymbol{\lambda}^{k,n+1}).$$

Thus, this choice for α is very natural. Moreover, this choice yields an inner product for imposition of the rigid body motion which is a compromise between the $\mathbb{H}^1$ and the $\mathbb{L}^2$ inner products, and given by $\rho_1 \tau_0 (\cdot,\cdot)_0 + \mu_1 (\nabla\cdot, \nabla\cdot)_0$ where $(\cdot,\cdot)_0$ denotes the usual $\mathbb{L}^2$ inner product. It recovers the $\mathbb{L}^2$ inner product in the limit $\delta t \to 0$. It is precisely this trick, that makes the present algorithm more efficient compared to the DLM method (including the stress formulation of the DLM method presented in [10]). Instead of solving one global problem for the extended velocity $\mathbf{u}_1$ [2, Eq. (62)] and one local problem per particle [2, Eq. (65)], in the present case it is necessary to solve only one global problem for $\boldsymbol{\lambda}$ (Eq. (36) below) at each iteration step for imposition of the rigid body constraint. The numerical results with the DLM version presented in [6] and the present method seem to indicate that this choice for α also accelerates the convergence of this iteration.

Finally, the weak formulation of Eq. (31) is given by

Find $\boldsymbol{\lambda}^{k+2,n+1} \in \mathbb{H}_0^1(\Omega)$ *such that*

$$\begin{aligned}
&\int_\Omega \left(1 + \frac{\rho_1}{\rho_{2,i} - \rho_1}\right)\left(\rho_1 \tau_0 \delta\boldsymbol{\lambda}^{k+2,n+1}\mathbf{v} + \mu_1 \nabla \delta\boldsymbol{\lambda}^{k+2,n+1} \nabla \mathbf{v}\right)\mathrm{d}\Omega \\
&\quad = -\sum_{i=1}^{n} \int_{\Omega_{2,i}} \rho_1 \tau_0 \left(\mathbf{u}_1^{k+1,n+1} - \mathbf{u}_2^{k+1,n+1}\right)\mathbf{v} - \sum_{i=1}^{n} \int_{\Omega_{2,i}} \mu_1 \nabla \left(\mathbf{u}_1^{k+1,n+1} - \mathbf{u}_2^{k+1,n+1}\right)\nabla \mathbf{v}\,\mathrm{d}\Omega \\
&\qquad + \mu_1 \sum_{i=1}^{n} \int_{\partial\Omega_{2,i}} \mathbf{n} \nabla \left(\mathbf{u}_1^{k+1,n+1} - \mathbf{u}_2^{k+1,n+1}\right)\mathbf{v}\,\mathrm{d}s \quad \forall \mathbf{v} \in \mathbb{H}_0^1(\Omega).
\end{aligned} \tag{36}$$

The surface integral in the right-hand side is a result of the integration by parts of the Laplacian over each of the particles. The numerical experience shows that this integral as well as the surface integral in the first equation of (30) are usually very small and do not seem to alter significantly the results in case of spherical particles. Therefore, they are not taken into account in the results of Section 3.

The formulation given by Eqs. (33)–(36) are discretized by means of $\mathbb{P}_2 - P_1$ finite elements using $\mathbb{P}_2$ interpolation for the velocity and $\boldsymbol{\lambda}$ and P_1 interpolation for the pressure. The grid is a fixed tetrahedral Eulerian grid that discretizes Ω. The resulting linear systems are solved by means of a parallel version of the conjugate gradients method. To complete the discussion on the numerical algorithm, we need to specify the quadrature rule that is used to compute the integrals in these formulations. These are of course a Gauss type quadratures that insure the exactness of the integration. Only the integrals in the right hand side of Eq. (36) can cause some troubles since the domain of integration is not (in general) exactly covered with finite elements. In case that the surface of the particle intersects the interior of a given element we adjusted the Gaussian integration weights of the Gauss points inside the particle so that their sum is equal to one. This is a simple procedure that is not exact anymore but makes the corresponding quadrature consistent. More sophisticated solutions like adaptive integration procedures [11] can also be used but this simple fix seems to

work well on reasonably fine grids (see the next section). The ultimate solution of this problem is to subdivide the elements intersected by the boundaries of the particles into tetrahedra, so that the particle boundary is covered entirely with faces of these tetrahedra, and then integrate over this locally refined grid. This subdivision problem is relatively easy to perform algorithmically since the particle shape is known. Moreover, it does not require a lot of computational resources. We are presently working on the implementation of this approach.

2.5. Collision strategy

For handling more than one particle a collision treatment mechanism has to be added to prevent particles from interpenetrating each other. To avoid this overlapping a repulsive force was added to Eq. (16). Two different repulsive forces were tried. The first one was of the form suggested by Glowinski et al. [1], where a short-range repulsion force between particles that are near contact is introduced. For two given particles i and j the force is calculated as:

$$F_R = \frac{c_{i,j}}{\epsilon} \frac{(d_{i,j} - R_i - R_j - \rho)}{\rho} \frac{G_i G_j}{d_{i,j}}, \tag{37}$$

where $c_{i,j}$ is a scaling factor, ϵ is a small positive number, $d_{i,j}$ is the distance between the center of the two particles, R_i, R_j are the radius of the the particles, ρ is the range of the repulsion force and G_i, G_j are the center of mass of both particles. In this approach, the choice of the scaling and stiffness parameters ($c_{i,j}$ and ϵ) is very important, and in general, the ideal values of these parameters may vary. Therefore, we considered the following, nonparametric approach. We first check if the separation distance between the particles is larger than a given threshold value calculated as a function of the particles radius and the mesh resolution. If the distance is less than this value then the repulsive force is calculated iteratively so that both particles move along the line that passes through the center of mass of both particles and that the minimum distance is still maintained.

The algorithm can be summarized as follows (see also Fig. 1):

1. Estimate the particle's position $\mathbf{X}_i$ with the velocity calculated from Eq. (32).
2. Calculate the separation distance between particles and detect possible collisions:

$$s_{i,j} = |\mathbf{X}_i - \mathbf{X}_j| - (R_i + R_j), \tag{38}$$

   where $s_{i,j}$ is the separation distance between two spherical particles i, j, $\mathbf{X}_i$ and $\mathbf{X}_j$ are the centroidal coordinates of the ith and jth particles and R_i and R_j are their radii.
3. Update the particles positions. If $s_{i,j}$ is less than a minimum separation distance (the maximum allowed thickness of the film between any two particles) ϵ then the ith (respectively, jth) particle is moved away at a distance $\Delta\mathbf{r}_i = (M_j(\epsilon - s_{i,j}))/(M_i + M_j)$ (respectively $\Delta\mathbf{r}_j = (M_i(\epsilon - s_{i,j}))/(M_i + M_j)$), alongside the line connecting the centroids.
4. Step 3 is equivalent to a correction of the velocity of the ith particle given by

$$\Delta\mathbf{U}_i = \frac{\Delta\mathbf{r}_i}{\delta t}. \tag{39}$$

This correction is added to the particle's velocity. On the other hand, this correction is equivalent to adding of an additional force to the particle momentum equation.

In order to impose that the fluid velocity in the region occupied by the particles is equal to the new particle velocity we need to perform again *Substep 4* of the splitting algorithm above and continue iteratively until both, the rigid body constraint and the minimum separation distance criteria are met. However, if we choose ϵ to be small, one iteration of this algorithm is enough to impose them with a reasonable

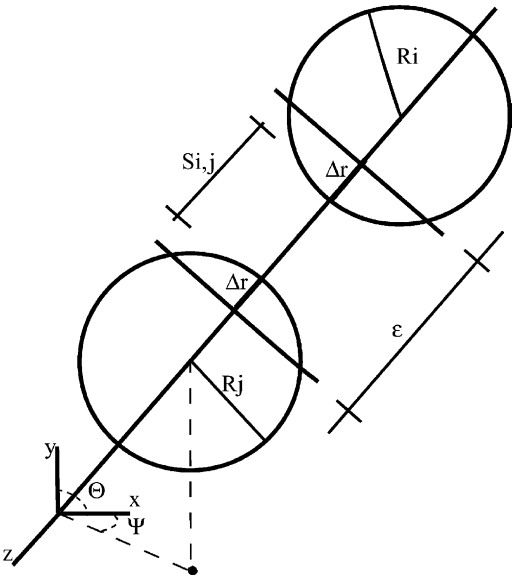


Fig. 1. Two particles near collision.

accuracy. Note that the maximum film thickness should be of the order of the mesh size in this area because otherwise the accuracy of the velocity in the film is doubtful. Also, this collision prevention mechanism works well only in the case of spherical particles and needs proper generalization for other particle shapes.

It is well known that before two particles touch each other, the thin film formed between them should drain completely. This is a physical phenomenon of a (time and spatial) scale different from the scale of the macromotion described by the momentum equations given above. Therefore, a more careful approach would require the development of a separate model for the film drainage and the incorporation of this model into the macromodel considered in this paper.

3. Numerical results

3.1. A performance test

The efficiency of the present algorithm (A1) was compared to the efficiency of the algorithm described in [6] (A2) on a test case for the sedimentation of several solid particles. As it has been already mentioned, A2 is a second order version of the DLM method proposed in [2]. Both codes were tested on a single processor Pentium 4 machine. The parametric setting was kept the same for both algorithms ($Re = 41$ and $Fr = 0.3$), and the mesh contained 179,401 nodes, 120,000 elements. The only difference between the test cases was the number of solid spheres which was set to 2, 4 and 6 correspondingly. The two codes had the same stopping criterion for the iteration which imposes the rigid body motion and for the conjugate gradient linear solver. The rigid body iteration was accepted as satisfactory converged if the maximum ratio of the correction to the particle velocity and the particle velocity itself was less then 1%. After the initial two steps when it required many steps to converge, the algorithm A2 needed five rigid body iterations per time step, independently of the number of spheres. Within each such iteration it needed to invert the linear system for the fluid velocity which required, at the average, slightly more than nine (conjugate gradient) iterations for each

component of the velocity. For the same problem, the present algorithm (A1) converged within one rigid body iteration per time step. The number of (conjugate gradient) iterations for each component of the velocity was between seven and eight (with the same stopping criterion). Taking into account that the algorithm A2 requires, in addition to the solution of the system for the fluid velocity (which is also solved by A1), to solve also one (relatively) small linear system for the Lagrange multiplier in each particle, it is clear that the present algorithm is faster. Moreover, the saving in CPU time increases with the number of spheres involved. The decrease in the number of rigid body iterations should be attributed to the inner product used to impose the rigid body constraint in the present case.

Since it is natural to think that the decrease in the number of iterations would lead to loss of accuracy, the accuracy of the method was further tested on three problems for which either experimental data is available or which have a well-known solution.

3.2. Case 1: single particle sedimentation

The first benchmark problem was the sedimentation of a single particle in a Newtonian liquid. Mordant et al. [12] provide extensive experimental data for this flow at a variety of Reynolds numbers and density ratios. Fig. 2 shows the results obtained for a sphere of radius = 0.5 settling in a rectangular channel with $Re = 41$ and $Fr = 1.12$. The channel width is four times the diameter of the sphere. The Eulerian mesh has 179,401 nodes and 120,000 elements. The time step is set to 0.01. The particle accelerates immediately after it is released but eventually reaches an approximately constant (terminal) sedimentation velocity when the gravity is balanced by the drag force. In Fig. 3 the velocity obtained with the present method is compared to

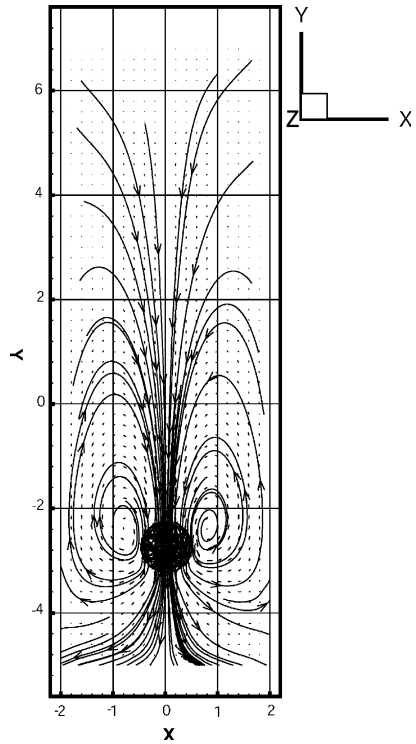


Fig. 2. Particle falling under gravity. $Re = 41$, $Fr = 1.12$.

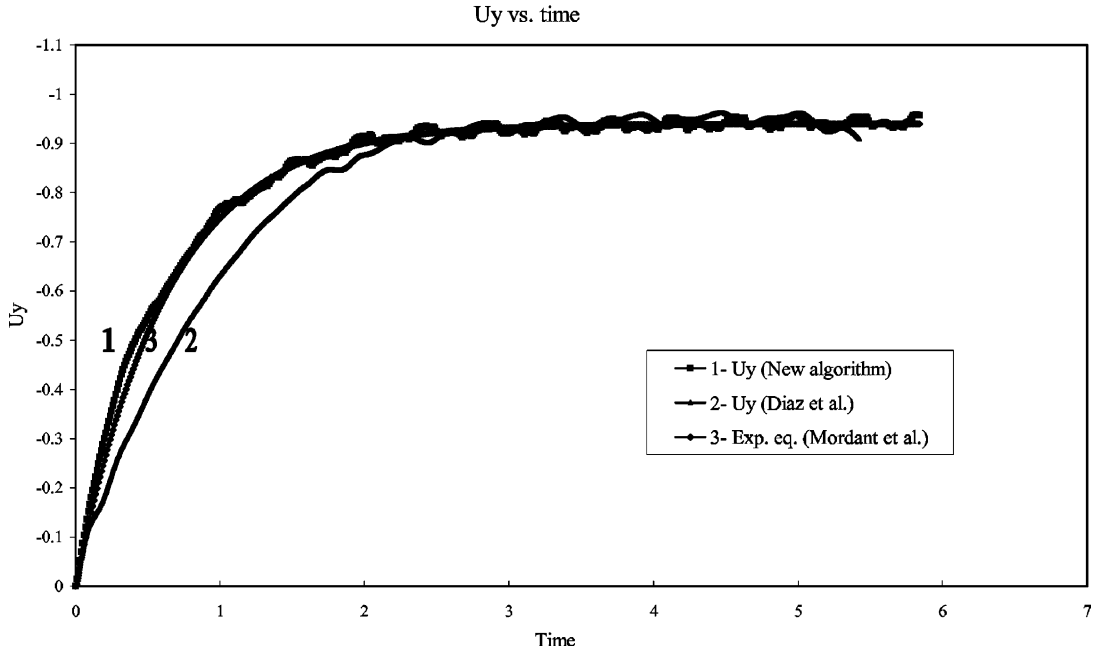


Fig. 3. Particle falling under gravity, comparison of experimental and numerical results. Vertical velocity vs. time.

that obtained with our (second order in time and space) implementation of the DLM method, and to the experimental measurement of Mordant et al. [12]. The latter authors experimentally studied the motion of a solid sphere settling under gravity in a fluid at rest. All their curves collapsed onto a single exponential curve given by

$$\frac{v_{\text{exp}}}{V_{\text{Terminal}}} = 1 - \exp\left(-\frac{3t}{\tau_{95}}\right),$$

where v_{exp} is the vertical velocity of the particle, V_{Terminal} is the terminal velocity, t is the time, and τ_{95} is the time it takes the particle to reach 95% of the terminal speed. The shape of the present curve matches closely the one produced with the equation proposed by Mordant [12] and the one in [6]. For this comparison the code presented in [6] used an auxiliary mesh for the sphere with 76 nodes and 226 elements. The mesh and time step used by both codes were the same. Fig. 4 shows the streamlines for the case of a sphere settling at a higher Reynolds number ($Re = 118.0$ and $Fr = 0.1$). At this Reynolds number a pair of vortices can be clearly seen behind the sphere.

3.3. Case 2: particle in a rotating cylinder

In order to test the angular velocity computation, simulations were carried out for the case of a neutrally buoyant spherical particle inside a cylinder of a radius 2 and and a height 4. The mesh in this case contained 63,679 nodes and 44,362 elements, and the time step was $\delta t = 0.01$. At the beginning of the simulation the fluid was at rest. The boundary condition at the wall of the cylinder imposes a rotation with an angular velocity $\Omega_z = 0.01$. The steady solution to this problem (at least at low Reynolds numbers) is a rigid body rotation inside the entire cylinder (including the particle) with an angular velocity $\Omega_z = 0.01$. Fig. 5 shows

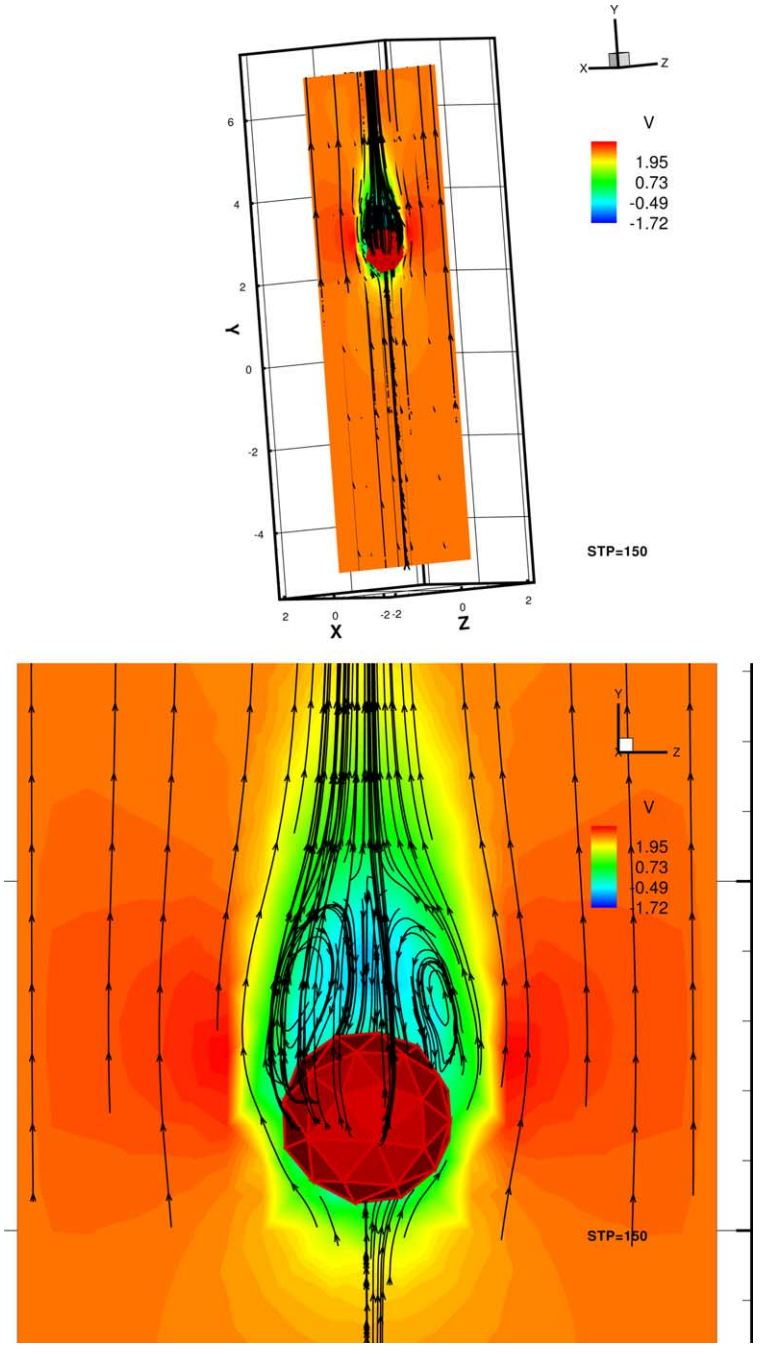
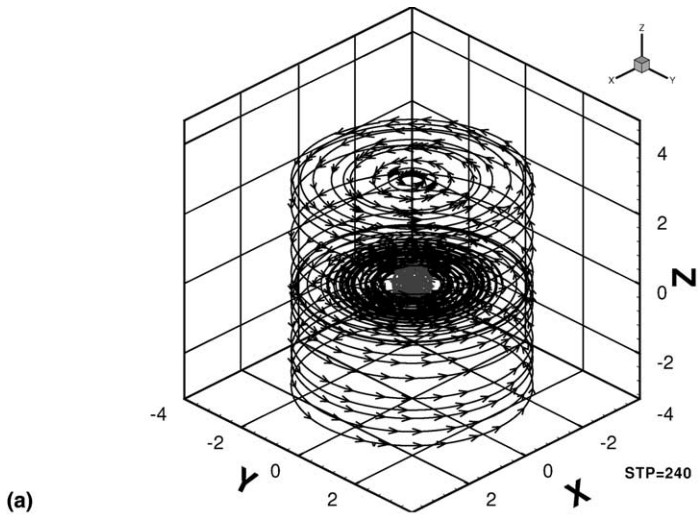


Fig. 4. Particle falling under gravity. $Re = 118$, $Fr = 0.10$.

the rotating cylinder and the evolution of the sphere's angular velocity with time. The angular velocity increases until it almost matches that of the wall. The difference between the angular velocities (of the sphere and the wall) can be attributed to the mesh resolution.



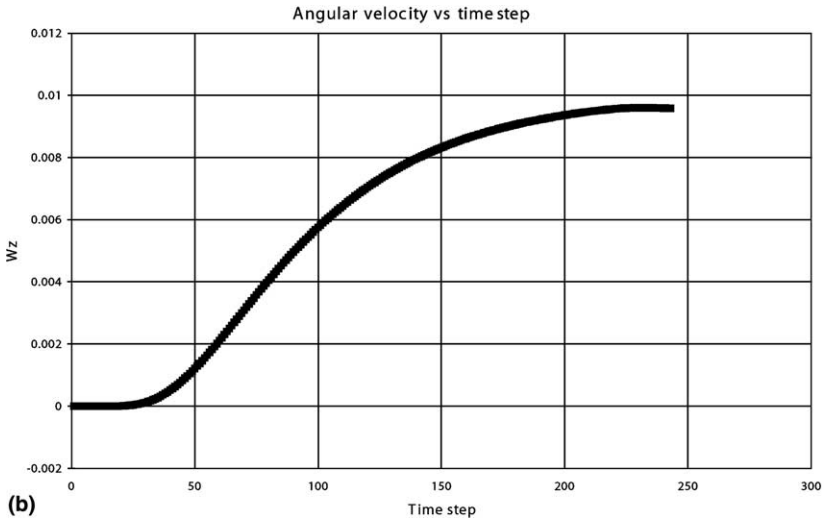


Fig. 5. (a) Sphere in a rotating cylinder. (b) Angular velocity vs. time (dimensionless).

3.4. Case 3: particle–particle interactions

To test the collision prevention mechanism, simulations were conducted using a mesh of 306,231 nodes and 205,800 elements and two spherical particles of an equal size. The Reynolds number was set to 67. The density ratio was $\rho_{2,i}/\rho_1 = 9.44$. Initially both, the liquid and the particles were at rest. No-slip conditions were prescribed on all the walls. Both spheres had a radius of 0.4. Their initial positions were $(0.0, 6.0, 0.0)$ and $(0.0, 5.0, 0.0)$, respectively.

Fig. 6 shows different frames captured with a high speed camera in our laboratory, the conditions of the experiment being similar to the conditions of the numerical simulation. Fig. 7 shows the results of our numerical simulations as described before. In both cases, the upper sphere initially settles slower than the bottom sphere. After a while, the velocity of this upper sphere increases and it starts catching the leading

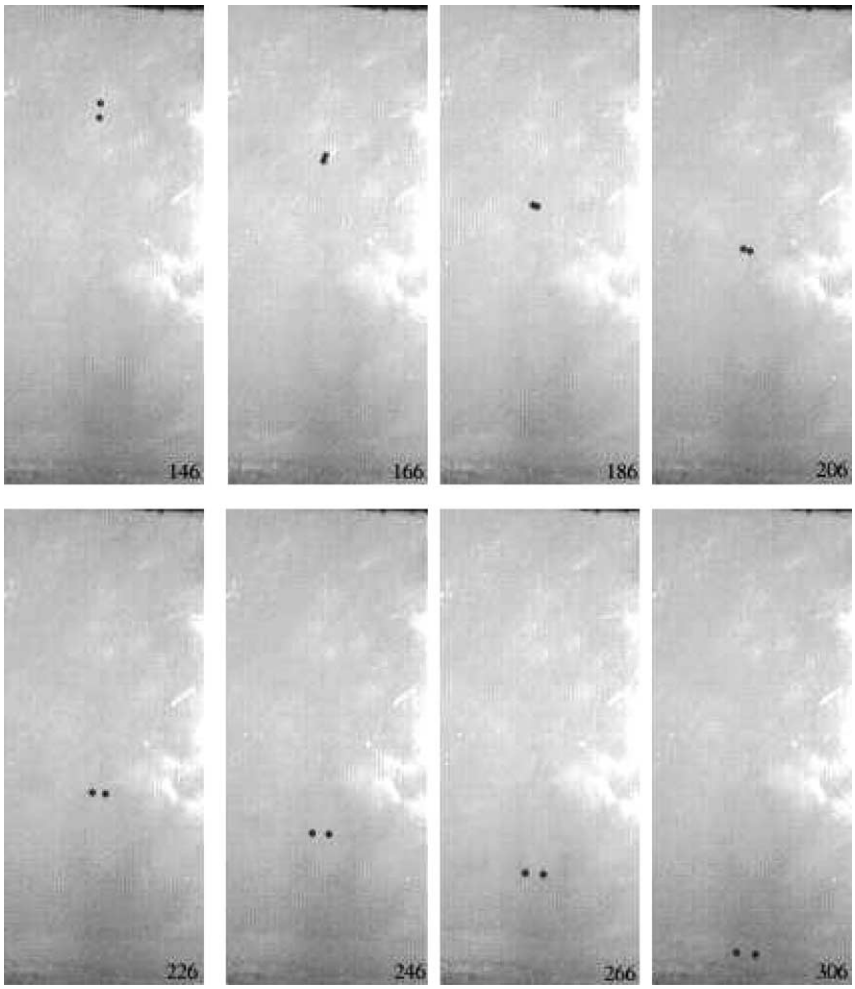


Fig. 6. Two spheres settling under gravity. Experimental results.

sphere. This is caused by the reduced drag that the upper sphere experiences when it enters the leading sphere's wake. The two spheres then kiss and continue dropping together. This new configuration is unstable and the pair tumbles. The spheres start to separate until a stable distance is achieved. Finally the two spheres fall side by side. Qualitatively, the result of the numerical simulation clearly matches the experimental results.

4. Conclusions

The method proposed in the present paper is a fictitious domain Eulerian method for direct simulation of particulate flows. The main advantages in comparison to other methods of the same type (in particular the method that we used as a starting point for the present study, proposed by Glowinski et al. [1]) are that:

1. It defines a global Lagrange multiplier, λ, whose physical meaning is of an additional velocity field that imposes the rigid body motion. This allows us to completely eliminate the Lagrangian grids used by [1] to

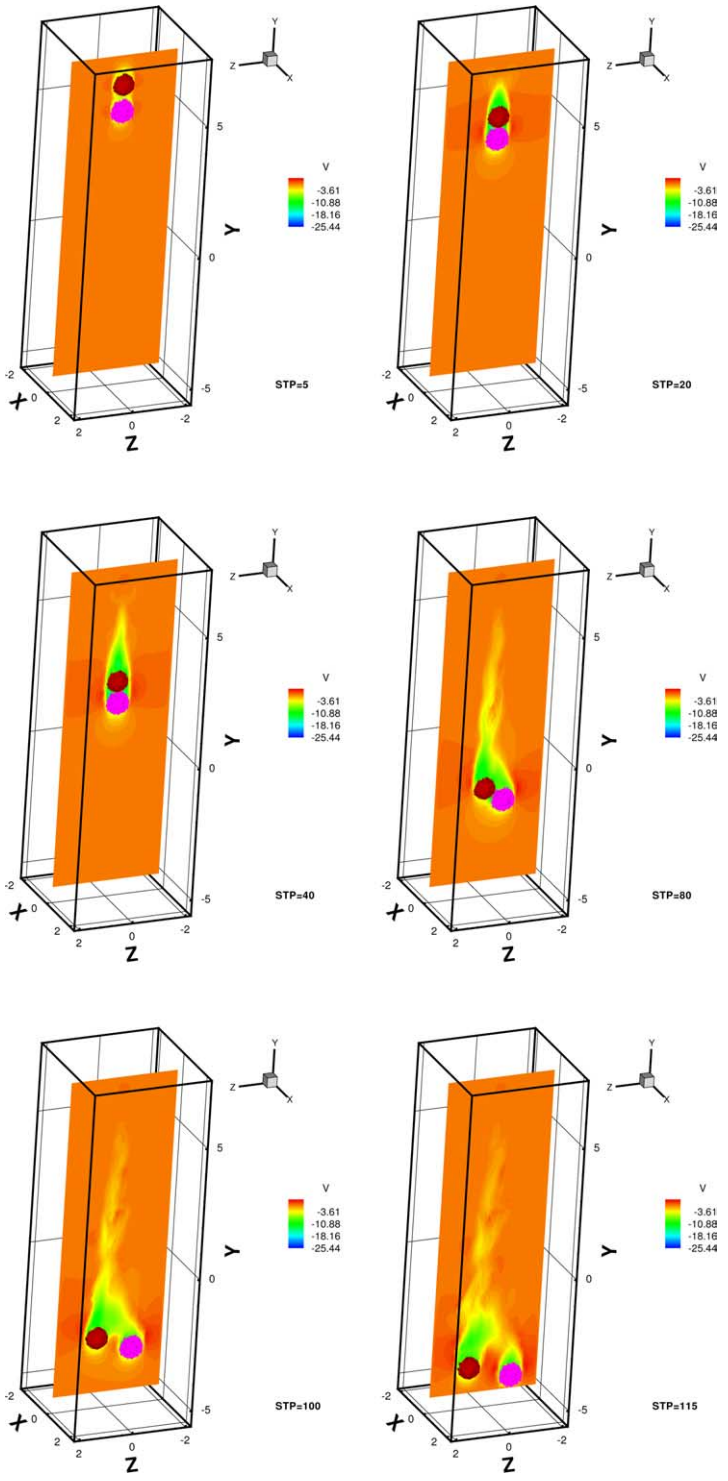


Fig. 7. Two spheres settling under gravity $t = 0.05$,0.20, 0.40, 0.80, 1.0, 1.15 (dimensionless).

discretize the distributed Lagrange multipliers in their case. This allows for the use of basically only one solver on relatively regular grids which facilitates the parallel implementation of the method.
2. The global Lagrange multiplier, together with a proper choice for the inner product that is used to impose the rigid body constraint allows to avoid the need to solve the linear system for computing the distributed Lagrange multipliers in each particle. Although this system is not large, it eventually is to be solved on each iteration of step of *Substep 4* of the algorithm above. If the flow contains many particles, the savings can be significant.
3. The present algorithm employs a second order in time, incremental projection scheme for the resolution of the generalized Stokes problem which, as our numerical experience shows, performs significantly better than a first order scheme for almost the same expenses. The spatial discretization is also second order accurate for the velocity and the Lagrange multiplier and first order accurate for the pressure.

In addition to these advantages, this method retains all the advantages of the DLM method proposed by Glowinski et al. [1] (see the conclusions of this paper).

Acknowledgements

The authors would like to thank Ahmed Sameh from Purdue University for the very fruitful discussions about the fictitious domain method. This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Alberta Informatics Circle of Research Excellence (iCORE).

References

[1] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow, J. Comput. Phys. 169 (2001) 363.
[2] R. Glowinski, T. Pan, T. Hesla, D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 25 (1999) 755.
[3] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for viscous incompressible multi-fluid flows, J. Comp. Physics 169 (2001) 708.
[4] H.H. Hu, Direct simulation of flows of solid–liquid mixtures, Int. J. Multiphase Flow 22 (1996) 335.
[5] A. Johnson, T. Tezduyar, 3D simulation of fluid–particle interactions with the number of particles reaching 100, Comp. Meth. Appl. Mech. Engng. 145 (1997) 301.
[6] C. Diaz-Goano, P. Minev, K. Nandakumar, A Lagrange multipliers/fictitious domain approach to particulate flows, in: W. Margenov, Yalamov (Eds.), Lecture Notes in Computer Science, vol. 2179, Springer, Sozopol, Bulgaria, 2001, p. 409.
[7] J. Guermond, J. Shen, Velocity-correction projection methods for incompressible flows, SIAM J. Num. Anal. 41 (2003) 112.
[8] P. Minev, C. Ethier, A semi-implicit projection algorithm for the Navier–Stokes equations with application to flows in complex geometries., in: M.G. et al. (Ed.), Notes on Numerical Fluid Mechanics, Vieweg, Braunschweig/Wiesbaden, 1999, p. 223, 73.
[9] J. Guermond, P. Minev, Analysis of a projection/characteristic scheme for incompressible flow, Comm. Numer. Meth. Engng. 19 (2003) 535.
[10] N. Patankar, P. Singh, D. Joseph, R. Glowinski, T. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 26 (2000) 1509.
[11] T. Chen, P. Minev, K. Nandakumar, A finite element technique for multifluid incompressible flow using Eulerian grids, J. Comp. Phys. 187 (2003) 255.
[12] N. Mordant, J. Pinton, Velocity measurement of a settling sphere, Eur. Phys. J. B 18 (2000) 343.