Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 867–879
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fictitious domain formulation for flows with rigid particles: A non-Lagrange multiplier version

C. Veeramani [a], P.D. Minev [b,*], K. Nandakumar [a]

[a] *Department of Chemical and Materials Engineering, University of Alberta, 677 Central Academic Building, Edmonton, Alberta, T6G 2G6 Canada*
[b] *Department of Mathematical and Statistical Sciences, University of Alberta, 677 Central Academic Building, Edmonton, Alberta, T6G 2G1 Canada*

Received 28 April 2006; received in revised form 19 October 2006; accepted 23 October 2006
Available online 8 December 2006

Abstract

In this paper, we present a development of the fictitious domain method proposed in Ref. [C. Diaz-Goano, P. Minev, K. Nandakumar, A fictitious domain/finite element method for particulate flows, J. Comput. Phys. 192 (2003) 105]. The main new feature of the modified method is that after a proper splitting, it avoids the need to use Lagrange multipliers for imposition of the rigid body motion and instead, it resolves the interaction force between the two phases explicitly. Then, the end-of-step fluid velocity is a solution of an integral equation. The most straightforward way to resolve it is via an iteration but a direct extrapolation is also possible. If the latter approach is applied then the fictitious domain formulation becomes fully explicit with respect to the rigid body constraint and therefore, the corresponding numerical procedure is much cheaper. Most of the numerical results presented in this article are obtained with such an explicit formulation.

MSC: 65N35; 65N22; 65F05

Keywords: Fictitious domain method; Particulate flow

1. Introduction

In Ref. [1] we proposed a Lagrange multiplier based fictitious domain method which used a global (non-distributed) Lagrange multiplier to enforce the rigid body constraint. It is a development of the distributed Lagrange multiplier method of Glowinski and co-workers (see Refs. [2–6]), the major difference being that the Lagrange multiplier was approximated in the same way as the fluid velocity field and that the rigid body constraint is enforced in an weighted H^1 norm. In the present paper we show that using the same starting

* Corresponding author. Tel.: +1 780 492 3398; fax: +1 780 492 6826.
E-mail addresses: veeramani@ualberta.ca (C. Veeramani), minev@ualberta.ca (P.D. Minev), kumar.nandakumar@ualberta.ca (K. Nandakumar).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.10.028

fictitious domain formulation it is possible to explicitly impose the rigid body motion on the discretized in time and split original set of PDEs and therefore we can get rid of the Lagrange multiplier. The resulting equation for the end-of-step velocity is an integral equation which can be resolved either directly (which is in general quite expensive) or iteratively. Since the integral that comes in the equation depends only on the curl of the fluid velocity in the domain occupied by the particles, the iterative method usually converges very fast. The method can also be made fully explicit by approximating the angular velocity of the particle from the previous time steps.

Non-Lagrange multiplier fictitious domain methods have been used for a long time. The most prominent example probably is the immersed boundary method of Peskin (see, for example [7]). It uses Dirac δ functions to enforce certain behaviour on boundaries immersed in a fluid. In the present work, the rigid body motion in the domain occupied by the particles gives rise to Heaviside functions in the right hand side of the momentum equations for the fluid. Unlike the δ function, if the problem is solved by a Galerkin method, it is not necessary to fit the grid to the boundary of the particles in order to obtain an optimal approximation. A scheme similar to the present one, has been proposed in Ref. [8,9]. It also eliminates the distributed Lagrange multiplier from the formulation, but there are some differences between the two schemes which will be highlighted in the next section.

The present method is validated on a number of problems involving rigid particles for which experimental data exists: a settling of a single particle in a container, and a single particle in a 3D Poiseuille's flow. Finally, results for the settling of an array of many particles in a container are shown.

2. Fictitious domain formulation and discretization

2.1. Formulation

Let us suppose that the fluid occupies a domain Ω_1 and has a constant density ρ_1 and a viscosity μ_1. Suppose also that within the fluid there are n rigid particles with constant densities $\rho_{2,i}$ which occupy a domain $\Omega_2 = \cup_{i=1}^{n} \Omega_{2,i}$ (the non-constant density case can be easily considered by a generalization of the formulation below). Let us 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 momentum equations for the fluid are the Navier–Stokes equations:

$$\frac{\mathrm{D}\hat{\mathbf{u}}_1}{\mathrm{D}t} = -\nabla \hat{p}_1 + \frac{1}{Re}\nabla^2 \hat{\mathbf{u}}_1, \quad \nabla \cdot \hat{\mathbf{u}}_1 = 0 \quad \text{in } \Omega_1. \tag{1}$$

Note that the gravity term is absorbed into the pressure gradient and the Reynolds number Re is defined using the fluid density and viscosity ρ_1 and μ_1. The equations of motion of the ith rigid particle are given by

$$\rho_{2,i}\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = (\rho_{2,i} - \rho_1)\frac{1}{Fr}\mathbf{e}_\mathrm{g} + \frac{\rho_1}{V_i}\mathbf{F}_i, \tag{2}$$

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

Here Fr is the Froude number, $\mathbf{e}_\mathrm{g}$ is the unit vector in the direction of gravity, V_i is the volume of the particle, $\mathbf{F}_i = \int_{\partial\Omega_{2,i}} \hat{\boldsymbol{\sigma}}_1 \mathbf{n} \mathrm{d}s$ is the total hydrodynamic force acting on the particle, $\hat{\boldsymbol{\sigma}}_1 = -\hat{p}_1\boldsymbol{\delta} + 1/Re(\nabla\hat{\mathbf{u}}_1 + (\nabla\hat{\mathbf{u}}_1)^\mathrm{T})$ is the stress tensor of the fluid, $\mathbf{n}$ is the unit normal pointing out of the particle, $\mathbf{I}_i$ is its tensor of inertia, and $\mathbf{T}_i$ is the hydrodynamic torque about its center of mass. Note, that the factor ρ_1 of $\mathbf{F}_i$ comes from the non-dimensionalization of the equation and that the similar factor in the angular velocity equation is absorbed into the tensor of inertia.

The key idea of the fictitious domain approach is to extend the equations of the fluid into Ω_2 so that we can discretize them without remeshing Ω_1 each time when the particles move which is the major disadvantage of the ALE methods. This would allow to resolve the flow using a relatively simple (in most cases structured) grid which allows for an easy parallelization. This approach also significantly simplifies the implementation of the method. The Lagrange multiplier methods further impose the rigid body motion as a side constraint using Lagrange multipliers. Ref. [2] suggests a method based on a Lagrange multiplier distributed on each particle, which is discretized by means of separate grids covering each of the particles. They also use an unified vari-

ational starting formulation for the fluid particle mixture (see Ref. [2] for details). In Ref. [1] we proposed to use a global Lagrange multiplier which is approximated in the same space where the fluid velocity is approximated. We also derived a set of fictitious domain PDEs/ODEs that govern the fluid and the particles. Here, we start with the same idea but instead of using a Lagrange multiplier to impose the rigid body constraint we derive an explicit equation for the interaction force between the two phases. This equation can be explicitly resolved and so, the interaction force can be eliminated from the system. Thus, we start, similarly to the derivation in Ref. [1], by extending the stress $\hat{\boldsymbol{\sigma}}_1 = -\hat{p}_1\boldsymbol{\delta} + 1/Re(\nabla\hat{\mathbf{u}}_1 + (\nabla\hat{\mathbf{u}}_1)^{\mathrm{T}})$ to the entire domain Ω. Let us denote the extension by $\boldsymbol{\sigma}_1 = -p_1\boldsymbol{\delta} + 1/Re(\nabla\mathbf{u}_1 + (\nabla\mathbf{u}_1)^{\mathrm{T}})$ with $\mathbf{u}_1$ being the extension of $\hat{\mathbf{u}}_1$ to the entire domain Ω. Then an application of the Gauss theorem gives that the total dimensionless hydrodynamic force acting on the ith particle is

$$\mathbf{F}_i = \int_{\partial\Omega_{2,i}} \hat{\boldsymbol{\sigma}}_1\hat{\mathbf{n}}_i \mathrm{d}s = \int_{\Omega_{2,i}} \nabla\cdot\boldsymbol{\sigma}_1 \mathrm{d}\Omega, \tag{4}$$

where $\hat{n}_i$ is the outward normal to the particle surface. Then the momentum equation for this particle becomes:

$$\frac{\rho_{2,i}}{\rho_1}\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \frac{\rho_{2,i}-\rho_1}{\rho_1}\frac{1}{Fr}\mathbf{e}_{\mathrm{g}} + \frac{1}{V_i}\int_{\Omega_{2,i}} \nabla\cdot\boldsymbol{\sigma}_1 \mathrm{d}\Omega. \tag{5}$$

Similarly to Ref. [1] we define the interaction force,

$$\hat{\mathbf{F}} = \begin{cases} -\frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t} + \frac{1}{Re}\nabla^2\mathbf{u}_1 - \nabla p_1, & \text{in } \Omega_{2,i}, \quad i = 1,\ldots,n \\ 0, & \text{in } \Omega_1 \end{cases} \tag{6}$$

which allows us to extend the Navier–Stokes equations to the entire domain Ω as

$$\frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t} = -\nabla p_1 + \frac{1}{Re}\nabla^2\mathbf{u}_1 - \hat{\mathbf{F}}, \quad \nabla\times\mathbf{u}_1 = 0 \quad \text{in } \Omega. \tag{7}$$

Then the particle momentum equation can be written as

$$\frac{\rho_{2,i}}{\rho_1}\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} - \frac{1}{V_i}\int_{\Omega_{2,i}}\frac{\mathrm{D}}{\mathrm{D}t}\mathbf{u}_1\mathrm{d}\Omega = \frac{1}{V_i}\int_{\Omega_{2,i}}\left[\frac{\rho_{2,i}-\rho_1}{\rho_1}\frac{1}{Fr}\mathbf{e}_{\mathrm{g}} + \hat{\mathbf{F}}\right]\mathrm{d}\Omega. \tag{8}$$

Since the fluid in $\Omega_{2,i}$ should accelerate as if it is a rigid particle, we can impose the condition:

$$\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \frac{1}{V_i}\int_{\Omega_{2,i}}\frac{\mathrm{D}}{\mathrm{D}t}\mathbf{u}_1\mathrm{d}\Omega \tag{9}$$

and therefore rewrite the equation in the form,

$$\frac{\rho_{2,i}-\rho_1}{\rho_1}\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \frac{\rho_{2,i}-\rho_1}{\rho_1}\frac{1}{Fr}\mathbf{e}_{\mathrm{g}} + \frac{1}{V_i}\int_{\Omega_{2,i}}\hat{\mathbf{F}}\mathrm{d}\Omega. \tag{10}$$

In a fictitious domain formulation it is more convenient to redefine the interaction force $\hat{\mathbf{F}}$ by

$$\mathbf{F} = \begin{cases} \frac{1}{Fr}\mathbf{e}_{\mathrm{g}} + \frac{\rho_1}{\rho_{2,i}-\rho_1}\hat{\mathbf{F}}, & \text{in } \Omega_{2,i}, \quad i = 1,\ldots,n \\ 0, & \text{in } \Omega_1 \end{cases} \tag{11}$$

and rewrite the set of equations in the form

$$\frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t} = -\nabla p_1 + \frac{1}{Re}\nabla^2\mathbf{u}_1 + \frac{\rho_{2,i}-\rho_1}{\rho_1}(\mathbf{G}-\mathbf{F}), \quad \nabla\cdot\mathbf{u}_1 = 0 \quad \text{in } \Omega \tag{12}$$

$$\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \frac{1}{V_i}\int_{\Omega_{2,i}}\mathbf{F}\mathrm{d}\Omega \tag{13}$$

where

$$\mathbf{G} = \begin{cases} \frac{1}{Fr}\mathbf{e}_{\mathrm{g}}, & \text{in } \Omega_{2,i}, \quad i = 1,\ldots,n \\ 0, & \text{in } \Omega_1. \end{cases} \tag{14}$$

Note that unlike the momentum equation in Ref. [9] (their equation 21), there is no density jump in Eq. (12) which can lead to a bad conditioning of the pressure Poisson's equation in case of large density differences between the fluid and the particles.

As suggested in Ref. [1], the angular velocity of the ith particle, $\boldsymbol{\omega}_i$, can be computed from the no-slip boundary condition on the surface of the ith particle. It reads

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

Then clearly

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

which yields,

$$\boldsymbol{\omega}_i = \frac{1}{2V_i} \int_{\Omega_{2,i}} \nabla \times \mathbf{u}_1 \mathrm{d}\Omega, \quad i = 1, \ldots, n. \tag{15}$$

The interaction force is to be determined from the condition that enforces the rigid body motion in $\Omega_{2,i}$, i.e.

$$\mathbf{U}_i + \boldsymbol{\omega}_i \times (\mathbf{x} - \mathbf{X}_i) = \mathbf{u}_1, \quad \text{in } \Omega_{2,i}, \quad i = 1, \ldots, n. \tag{16}$$

Since this condition does not contain the interaction force itself, it is a side constraint for the two momentum equations and therefore it is natural to impose it via Lagrange multipliers. It turns out, however, that an explicit equation for the interaction force can be derived as shown below.

The final system of equations is given by Eqs. (12), (13), (15) and (16), the condition (9) which follows from (16), and the equation for the position of the center of mass of each particle which reads

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

2.2. Discretization

It is almost universally accepted now a days that such problems should be discretized in time using an operator splitting procedure. Here we adhere to the second-order pressure-correction version of the procedure described in Ref. [1]. In case of a single-phase incompressible flow the pressure correction scheme is second-order accurate in time as proved in Ref. [10]. In the present case only a formal second-order accuracy can be claimed because of the fictitious domain formulation of the problem.

- *Advection-diffusion substep*

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

$$\mathbf{X}_i^{p,n+1} = \mathbf{X}_i^{n-1} + 2\delta t \mathbf{U}_i^n, \tag{17}$$

  where δt is the time step. Then we solve for $\mathbf{u}_1^*$ from

$$\begin{aligned} &\tau_0 \mathbf{u}_1^* - \frac{1}{Re} \nabla^2 \mathbf{u}_1^* = -\tau_1 \tilde{\mathbf{u}}_1^n - \tau_2 \tilde{\mathbf{u}}_1^{n-1} - \nabla p_1^n + \frac{\rho_{2,i} - \rho_1}{\rho_1} \mathbf{G}, \quad \text{in } \Omega \\ &\mathbf{u}_1^* = 0 \quad \text{on } \partial\Omega \end{aligned} \tag{18}$$

  where $\tau_0 = 3/(2\delta t), \tau_1 = -2/\delta t, \tau_2 = 1/(2\delta t)$, and $\tilde{\mathbf{u}}_1^n$, $\tilde{\mathbf{u}}_1^{n-1}$ are the velocities from time levels n, $n-1$ which are advected alongside an approximation of the characteristics (see Ref. [11] for details). To compute them we first approximate the foots of the characteristic at any given point $\mathbf{x}(t^{n+1})$, denoted by $\underline{\mathbf{x}}(t^{n-i}), i = 0,1$, from the following discretization of the characteristic equation:

$$\underline{\mathbf{x}}(t^{n-i}) = \mathbf{x}(t^{n+1}) - (i+1)\delta t \mathbf{u}_e^{n+1}, \quad i = 0, 1 \tag{19}$$

where $\mathbf{u}_e^{n+1} = 2\mathbf{u}_1^n - \mathbf{u}_1^{n-1}$ is a second order extrapolation for the velocity at $\mathbf{x}(t^{n+1})$ at time t^{n+1}. Then the advected velocity is given by $\tilde{\mathbf{u}}_1^n(\mathbf{x}) = \mathbf{u}_1^n(\underline{\mathbf{x}}), \tilde{\mathbf{u}}_1^{n-1}(\mathbf{x}) = \mathbf{u}_1^{n-1}(\underline{\mathbf{x}})$.
Since at this substep the rigid body constraint is not taken into account (i.e. $\mathbf{F} = 0$) Eq. (13) would yield a zero guess for the particle velocities $\mathbf{U}_i$. A better prediction for them can be obtained if they are computed from Eq. (9) which can be obtained by differentiating the rigid body constraint (16) in time and integrating the result over $\Omega_{2,i}(t^{n+1})$. Its discretization is given by

$$\tau_0 \mathbf{U}_i^* + \tau_1 \mathbf{U}_i^n + \tau_2 \mathbf{U}_i^{n-1} = \frac{1}{V_i} \int_{\Omega_{2,i}(t^{n+1})} (\tau_0 \mathbf{u}_1^* + \tau_1 \tilde{\mathbf{u}}_1^n + \tau_2 \tilde{\mathbf{u}}_1^{n-1}) \mathrm{d}\Omega. \tag{20}$$

At the last substep, this prediction for $\mathbf{U}_i$ will be corrected using (13).

- *Projection substep*
  On the next substep we impose the incompressibility constraint by solving:

$$\begin{aligned} &\tau_0(\mathbf{u}_1^{**} - \mathbf{u}_1^*) = -\nabla(p_1^{n+1} - p_1^n) \quad \text{in } \Omega \\ &\nabla \cdot \mathbf{u}_1^{**} = 0 \quad \text{in } \Omega \\ &\mathbf{u}_1^{**} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{21}$$

  $\mathbf{n}$ being the outward normal to $\partial\Omega$.

- *Rigid body constraint*
  On the last substep we impose the rigid body motion in $\Omega_{2,i}$. In Refs. [4,1] this is done by means of an iterative procedure which is somewhat in contradiction with the spirit of the splitting procedure and makes the overall algorithm quite inefficient. Indeed, on the previous substep it was possible to derive an explicit equation for the pressure which is a Lagrange multiplier for the incompressibility constraint while on this substep, [4,1] compute the Lagrange multiplier iteratively. Below we show how to derive an equation for $\mathbf{F}$. Following the Marchuk/Yanenko splitting idea, the next substep should read:

$$\begin{aligned} &\tau_0(\mathbf{u}_1^{n+1} - \mathbf{u}_1^{**}) = -\frac{\rho_{2,i} - \rho_1}{\rho_1} \mathbf{F} \quad \text{in } \Omega, \\ &\tau_0\left(\mathbf{U}_i^{n+1} - \mathbf{U}_i^*\right) = \frac{1}{V_i} \int_{\Omega_{2,i}(t^{n+1})} \mathbf{F} \mathrm{d}\Omega, \\ &\mathbf{u}_1^{n+1} - \left(\mathbf{U}_i^{n+1} + \boldsymbol{\omega}_i^{n+1} \times (\mathbf{x} - \mathbf{X}_i^{p,n+1})\right) = 0 \quad \text{in } \Omega_{2,i}(t^{n+1}), \quad i = 1, \ldots, n. \end{aligned} \tag{22}$$

  Subtracting the first two equations and imposing the rigid body motion as formulated in the last equation, we obtain:

$$\begin{aligned} &-\frac{\rho_{2,i} - \rho_1}{\rho_1} \mathbf{F} - \frac{1}{V_i} \int_{\Omega_{2,i}(t^{n+1})} \mathbf{F} \mathrm{d}\Omega = \tau_0\left(\mathbf{U}_i^* - \mathbf{u}_1^{**}\right) + \frac{\tau_0}{2V_i} \left(\int_{\Omega_{2,i}(t^{n+1})} \nabla \times \mathbf{u}_1^{n+1} \mathrm{d}\Omega \right) \times (\mathbf{x} - \mathbf{X}_i^{p,n+1}) \quad \text{in } \Omega_{2,i}(t^{n+1}), \\ &i = 1, \ldots, n. \end{aligned} \tag{23}$$

  Integrating over $\Omega_{2,i}(t^{n+1})$ (note that the integral of the last term in Eq. (23) is equal to zero) we get:

$$\int_{\Omega_{2,i}(t^{n+1})} \mathbf{F} \mathrm{d}\Omega = \frac{\rho_1}{\rho_{2,i}} \tau_0 \int_{\Omega_{2,i}(t^{n+1})} \left(\mathbf{u}_1^{**} - \mathbf{U}_i^*\right) \mathrm{d}\Omega. \tag{24}$$

Using Eqs. (23) and (24) we obtain the following equation for the interaction force:

$$-\frac{\rho_{2,i}-\rho_1}{\rho_1}\mathbf{F}=\sum_{i=1}^{n}\left[\tau_0\left(\mathbf{U}_i^*-\mathbf{u}_1^{**}\right)+\frac{\tau_0}{2V_i}\left(\int_{\Omega_{2,i}(t^{n+1})}\nabla\times\mathbf{u}_1^{n+1}\mathrm{d}\Omega\right)\times\left(\mathbf{x}-\mathbf{X}_i^{p,n+1}\right)+\frac{\tau_0}{V_i}\frac{\rho_1}{\rho_{2,i}}\int_{\Omega_{2,i}(t^{n+1})}\left(\mathbf{u}_1^{**}-\mathbf{U}_i^*\right)\mathrm{d}\Omega\right]1_{\Omega_{2,i}},\tag{25}$$

where

$$1_{\Omega_{2,i}}=\begin{cases}1, & \text{in } \Omega_{2,i}\\ 0, & \text{in } \Omega\setminus\Omega_{2,i}.\end{cases}\tag{26}$$

is the characteristic function of $\Omega_{2,i}$. Substituting this expression into the first equation of Eqs. (22) and (24) into the second equation of (22) we finally obtain the following set of equations for $\mathbf{u}_1$ and $\mathbf{U}_i$, $i=1,\ldots,n$,

$$\mathbf{u}_1^{n+1}=\mathbf{u}_1^{**}+\sum_{i=1}^{n}\left[\left(\mathbf{U}_i^*-\mathbf{u}_1^{**}\right)+\frac{1}{2V_i}\left(\int_{\Omega_{2,i}(t^{n+1})}\nabla\times\mathbf{u}_1^{n+1}\mathrm{d}\Omega\right)\times\left(\mathbf{x}-\mathbf{X}_i^{p,n+1}\right)+\frac{1}{V_i}\frac{\rho_1}{\rho_{2,i}}\int_{\Omega_{2,i}(t^{n+1})}\left(\mathbf{u}_1^{**}-\mathbf{U}_i^*\right)\mathrm{d}\Omega\right]1_{\Omega_2}\quad\text{in }\Omega,$$

$$\mathbf{U}_i^{n+1}=\frac{1}{V_i}\frac{\rho_1}{\rho_{2,i}}\int_{\Omega_{2,i}(t^{n+1})}\mathbf{u}_1^{**}\mathrm{d}\Omega+\left(1-\frac{\rho_1}{\rho_{2,i}}\right)\mathbf{U}_i^*.\tag{27}$$

The first equation contains a non-local term in the right hand side which depends on $\mathbf{u}_1^{n+1}$ and therefore its spatial discretization would yield a non-sparse matrix. One way to avoid this problem is to approximate $\nabla\times\mathbf{u}_1^{n+1}$ by $\nabla\times\mathbf{u}_1^{**}$ and so to impose the rigid body constraint fully explicitly. The other possibility is to solve the equation iteratively.

Note that if instead of computing $\mathbf{U}_i^*$ from Ref. (20) we do it, as suggested by Ref. [9], from $\mathbf{U}_i^*=\int_{\Omega_i}\mathbf{u}_1^{**}$ and if $\nabla\times\mathbf{u}_1^{n+1}$ is extrapolated by $\nabla\times\mathbf{u}_1^{**}$ then Eq. (27) yields the same velocity in the rigid body domain as the one resulting from the explicit procedure suggested in Ref. [9]. However, Eq. (27) can also be used for an iteration to refine the computation of the rigid body/fluid velocities as well.

Finally, the position of each particle is corrected according to

$$\mathbf{X}_i^{n+1}=\mathbf{X}_i^n+0.5\delta t(\mathbf{U}_i^{n+1}+\mathbf{U}_i^n).\tag{28}$$

The set of Eqs. (18), (21) and (27) are discretized in space by means of P_2–P_1 tetrahedral finite elements (see [1] for details). The resulting linear systems are solved using a conjugate gradient solver. A parallelized version of the method has also been developed using the PETSc libraries.

The integrals over the domain occupied by the particles, which appear in Eqs. (15), (20) and (27) can be approximated by the usual Gaussian quadratures used in finite element algorithms. However, this approximation can be very inaccurate in some cases because the functions to be integrated can be discontinuous within the elements intersected by the particle surfaces. Therefore, we developed a procedure that subdivides such elements into sub-elements which are exactly aligned with the particle surfaces (more details on this procedure can be found in Ref. [12]). This new grid is used only for the computation of these integrals and the Gaussian quadrature on such a grid is exact. It can significantly improve the accuracy of the overall algorithm. The results on the particle migration in a channel which are presented below are computed with this improved integration procedure.

2.3. Collision mechanism

One of the most difficult problems in the numerical modelling of particulate flows is the modelling of the collision of a particle with other particles and the walls of the container. When the distance between the bodies becomes very small, a thin fluid film between them is formed and a new spatial scale related to the film thickness appears in the problem. In order to resolve this scale, a tremendous grid refinement (in space and time) is required which is impossible even using modern supercomputers. Therefore, the only available option is to use a subgrid modelling approach which results in some sort of a lubrication force. For the collision of a particle with a plane wall we used the lubrication force suggested by Ref. [13]:

$$F_i^w=\begin{cases}-6\pi r_i U_\perp\mu_1\left(\frac{r_i}{\hat{h}}-\frac{r_i}{h}\right), & \text{if } \hat{h}<h\\ 0, & \text{otherwise}\end{cases}\tag{29}$$

where r_i is the radius of the particle, $U_\perp$ is its velocity component perpendicular to the wall, h is the grid size, and $\hat{h}$ is the gap between the wall and the particle. This force is used to correct the velocity of the particle that is used to predict/correct the particle position in Eq. (17)/(28).

For the collision between two particles we used the procedure suggested by Ref. [1]. It can be summarized as follows:

1. Estimate the particle's position $\mathbf{X}_i$ with the velocity calculated from Eq. (17) or (28).
2. Calculate the separation distance between particles and detect possible collisions.

$$s_{i,j} = |\mathbf{X}_i - \mathbf{X}_j| - (r_i + r_j) \tag{30}$$

   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 $\Delta\mathbf{r}_i = \frac{M_j(\epsilon - s_{i,j})}{M_i + M_j}$ (respectively $\Delta\mathbf{r}_j = \frac{M_i(\epsilon - s_{i,j})}{M_i + M_j}$), alongside the line connecting the centroids.

These are quite primitive models and obviously there is a need for better subgrid modelling. One possible approach is to derive some sort of a solution (or an approximation to the solution) for the velocity of the film drainage problem and compute the interaction force from it.

2.4. Numerical results

In all simulations for sedimentation of particles presented below, the angular velocity at time level $n+1$ in (27) is extrapolated according to $\nabla \times \mathbf{u}_1^{n+1} = \nabla \times \mathbf{u}_1^{**}$. In all simulations for the migration of neutrally buoyant particles, the average of two simple iterations are performed with respect to the angular velocity of the particle.

The first problem that we considered is about the sedimentation of a single particle with a density $\rho_2 = 1120\ \mathrm{kg/m^3}$ in a container. It is well documented experimentally in Ref. [13]. The container has dimensionless size $7 \times 7 \times 11$ which is very close to the size of the container used in the experiment. We used two different structured grids: a coarser grid of 134139 velocity nodes with a grid size in the vicinity of the particle equal to 0.2, and a finer grid of 467261 velocity nodes with a grid size in the vicinity of the particle equal to 0.1 (see Fig. 1). The Reynolds number is based on the particle diameter and the terminal velocity u_∞ of such a particle measured in the experiments of Ref. [13]. Table 1 provides a detailed information about the parameters used in this test problem.

In Fig. 2 we show the results for the vertical component of the centroidal velocity of a settling particle for a variety of Reynolds numbers, using the finer grid and three different time steps: 0.05, 0.01 and 0.005. They are compared to the corresponding experimental results from Ref. [13]. The terminal velocities match the experimental data quite well. Also, there is a clear convergence towards the experimental results with the decrease of the time step. The deviation that occurs at small Reynolds numbers and large times is due to the fact that the numerical speed of the particle is consistently greater than the experimental data and this causes the particle to reach the bottom faster. Another reason is the expression (29) for the interaction force between the particle and the bottom wall of the container which is quite heuristic. Obviously, a more sophisticated model is needed and this is a subject of our future studies. At first glance these results seem to be worse than the simulation results obtained by Ref. [13] with a lattice-Boltzmann method. However, it should be taken into account that these authors used a calibration procedure which computes an effective sphere radius which is computed from an analytic expression for the drag force at low Reynolds numbers at a given volumetrically averaged fluid velocity. This is in fact a fitting procedure used to compensate the numerical error in unresolved simulations. Without this procedure, as indicated in Ref. [13], the velocity of the particle can be up to 20% different from the experimental values. Our simulations do not include any fitting parameters except for the lubrication force (29) which acts only at the very end of the simulations. They also clearly show the need for the use of a finer grid and smaller time step which require very heavy computations and therefore we did not attempt them.

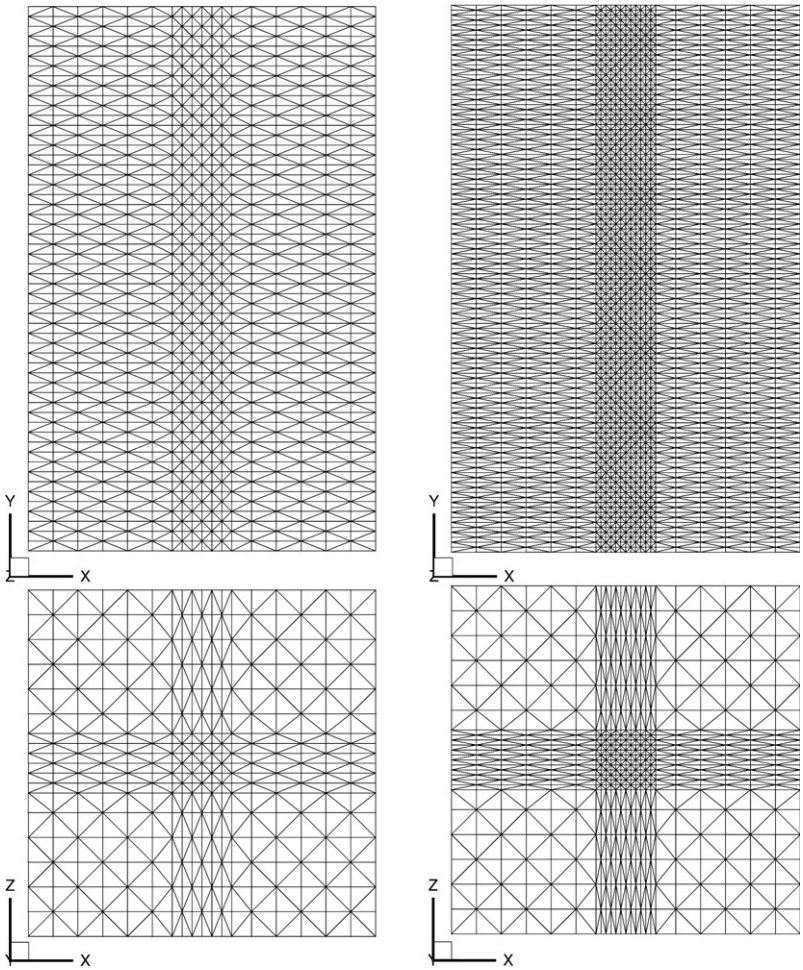


Fig. 1. Vertical (upper graphs) and horizontal cross sections (lower graphs) of the two structured grids used in the sedimentation simulations.

Table 1
Parameters for the sedimentation problem

ρ_1 (kg/m^3)	μ_1 (Ns/m^2)	u_∞ (m/s)	$Re = \frac{\rho_f u_\infty d_p}{\mu_f}$ (–)	$Fr = \frac{u_\infty^2}{g d_p}$ (–)
970.0	0.373	0.038	1.5	0.0098
965.0	0.212	0.060	4.1	0.0245
962.0	0.113	0.091	11.6	0.0563
960.0	0.058	0.128	31.9	0.1114

However, the convergence of the results with respect to the time step is obvious. In Fig. 3, we compare the computed centroidal velocities on the two different grids using a time step of 0.01. Obviously, the grid resolution improves the results only at relatively high Reynolds numbers and the error is still dominated by the time step.

Finally, the results obtained with the present algorithm which is direct and explicit in the sense of the rigid body constraint, are compared to the results for the same problem obtained with the iterative method proposed in Ref. [1] (see also [12]) in Fig. 4. The iterative method performs on an average between 2 and 11 iterations per time step (for an accuracy of 0.01 in a maximum norm, depending on the Reynolds number). Moreover, the results with the present method are somewhat better which is probably due to the better initial

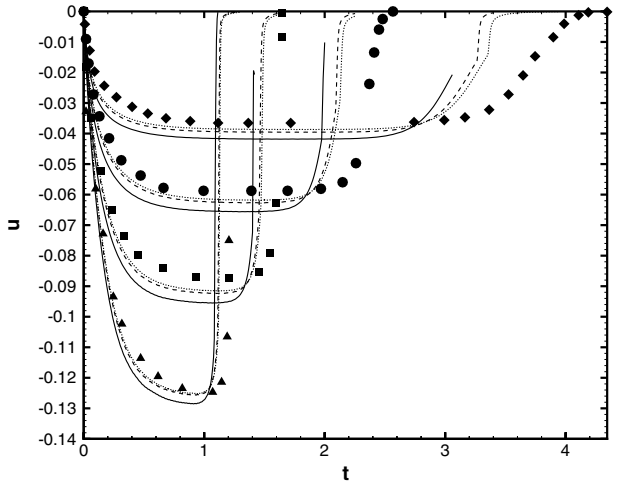


Fig. 2. Comparison of experimental (symbols) and numerical (lines) results for the vertical component of the particle velocity at four different Reynolds numbers: $Re = 1.5$ (◆), 4.1 (●), 11.6 (■), 31.9 (▲). $\delta t = 0.05$ (solid lines), $\delta t = 0.01$ (dashed lines) and $\delta t = 0.005$ (dotted lines). All results are produced on a fine grid of 467261 nodes.

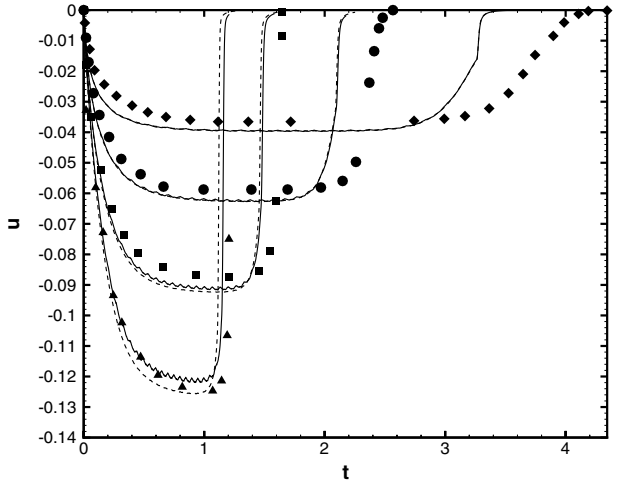


Fig. 3. Comparison for numerical results on fine mesh (467261 nodes, dashed lines) and coarse mesh (134139 nodes, solid lines) with experimental results (symbols) for the vertical component of the particle velocity at four different Reynolds numbers viz. $Re = 1.5$ (◆), 4.1 (●), 11.6 (■), 31.9 (▲). The numerical results are for time step of $\delta t = 0.01$.

guess for the end-of-step fluid velocity. The reason for the better guess is that in the present algorithm the gravity which is the dominant driving force, is transferred into the Navier–Stokes equations which significantly improves the initial guess for the particle velocity. Thus, the present method is much more efficient than the rigid body iteration suggested in Ref. [1].

In Fig. 5 we present snapshots from the simulation of the sedimentation of 64 equally sized spherical particles in a box of dimensions $6 \times 15 \times 6$. The Reynolds number (based on the diameter of the particles) is $Re = 10$, the Froude number is $Fr = 0.1$, and the ratio of the particle to fluid densities is $\rho_2/\rho_1 = 2.18$. We used a uniform grid containing 494371 velocity nodes and 73036 pressure nodes with a mesh size (the size of P_2–P_1 elements) of 0.2 in x, y and z directions. The time step was fixed to 0.01. The particle cloud clearly develops a Rayleigh–Taylor instability with the particles around the cross section of the container traveling faster than the peripheral particles. Then the initial square wave is dispersed and diffused. The wave evolution can be

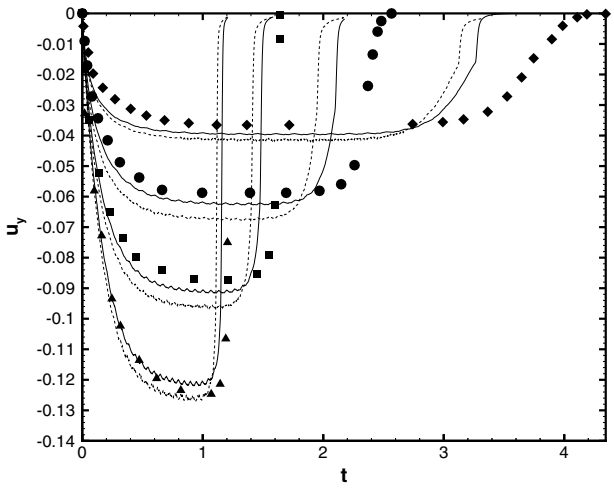
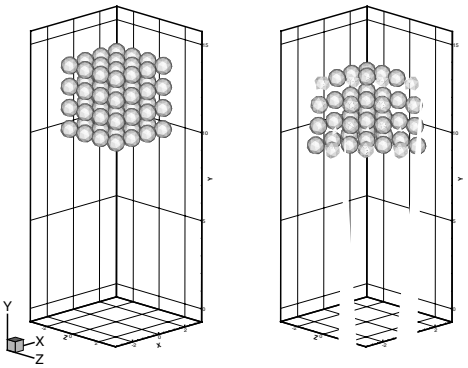


Fig. 4. Comparison the vertical component of the particle velocity of numerical (lines) and experimental (symbols) results on a coarse mesh (134139 nodes) using old iterative method (dashed lines) and new direct method (solid lines) with time step size $\delta t = 0.01$. Experimental results are at $Re = 1.5$ (◆), 4.1 (●), 11.6 (■), 31.9 (▲).



clearly seen in Fig. 6 where v pre ent the average velocity of the cloud of particles as well as the standard deviation of ıe pa ticles given by

$$\sqrt{\frac{\sum_{i=1}^{n}(x_c - x_i)^2}{n}}.$$

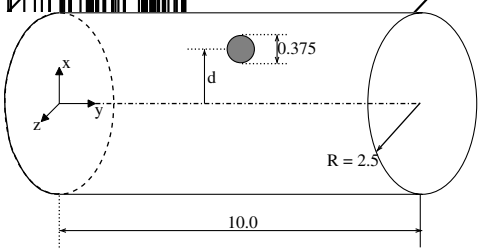


Fig. 7. A sketch of the channel geometry.

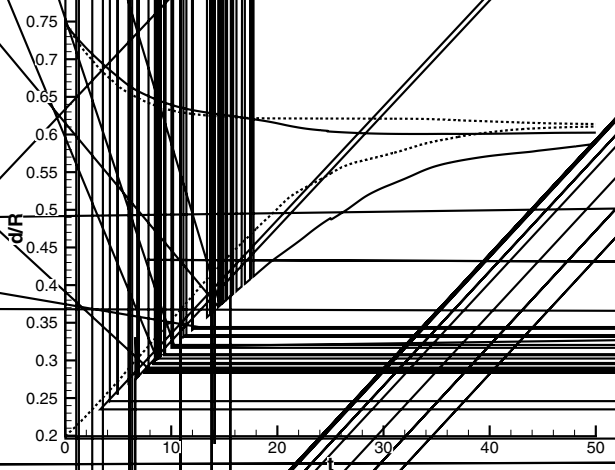


Fig. 8. The radial position of the particle as a function of time for starting positions 0.75 and 0.2. The lines indicate the present results on a grid of 1.07×10^6 nodes and $\delta t = 0.01$ (solid lines) or $\delta t = 0.005$ (dashed lines). The symbols indicate the results of Ref. [15].

Table 2
A comparison with the DLM and ALE results from Ref. [15,16] for the migration of a neutrally buoyant particle. The maximum channel velocity is 20 and the particle Reynolds number (based on the particle diameter) is 9.18

d/R	U_y			ω_z		
$U_m = 20.0$	DLM	ALE	Present	DLM	ALE	Present
0.10	19.4965	19.493	19.5805	0.7751	0.7875	0.7812
0.20	18.8841	18.881	18.9658	1.5514	1.5725	1.5734
0.30	17.8656	17.862	17.9730	2.3235	2.3539	2.3537
0.40	16.4442	16.439	16.5582	3.0872	3.1284	3.1523
0.50	14.6210	14.616	14.7561	3.8409	3.8918	3.9060
0.60	12.3957	12.388	12.5155	4.5824	4.6424	4.7483
0.70	9.7338	9.705	9.9333	5.2798	5.3295	5.3735
0.75	8.1822	8.127	8.3873	5.5765	5.5710	5.7214

ied this problem numerically using their version of the fictitious domain method. Here, we are comparing our results with theirs in case of a single particle. The geometry of the problem is sketched in Fig. 7. The viscosity of fluid is equal to 1 and the densities of the fluid and solid are also equal to 1. The steady values of the vertical component of the particle velocity U_y and the z-component of its angular velocity Ω_z are very close to the results obtained with the DLM method and the ALE results of Ref. [16] (see Table 2). The evolution towards the steady solutions is also quite similar as long as the starting position of the particle is close to the wall but deviates more from the DLM/ALE results when the starting position is close to the centre of the pipe (see Fig. 8). We used a grid of 1.07×10^6 velocity nodes and a P_2–P_1 approximation so the resolution is certainly comparable to the resolution of Ref. [15] (2.16×10^6 velocity nodes and an iso-P_2–P_1 approximation). The reason for the deviation seem to be the larger time steps that we used ($\delta t = 0.01, 0,005$). However, it is clear from Fig. 8 that our results converge to the results of Ref. [15] which are computed with a time step equal to 0.001. The use of such a time step would result in prohibitively expensive computations and therefore we did not perform them.

3. Conclusions

In the present article, we present a modified fictitious domain formulation which should be classified as a non-Lagrange multiplier formulation. It does not introduce explicitly a Lagrange multiplier but rather computes directly the interaction force between the two phases which yields a non-local term in the Navier–Stokes equations. In this term only the *curl* of the fluid velocity is at the upper time level and it can be extrapolated using the velocity at the previous substep of the splitting algorithm. As a result, we can avoid the potentially costly iterations for imposition of the rigid body constraint and the method becomes much more efficient. The accuracy of the method is verified on two difficult flows involving rigid balls. The results compare very well to available experimental and numerical data.

Acknowledgments

This study has been supported by Discovery Grants of the National Science and Engineering Research Council of Canada (NSERC).

References

[1] C. Diaz-Goano, P. Minev, K. Nandakumar, A fictitious domain/finite element method for particulate flows, J. Comput. Phys. 192 (2003) 105.
[2] R. Glowinski, T. Pan, J. Periaux, Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies, Comp. Meth. Appl. Mech. Eng. 151 (1998) 181.
[3] 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.
[4] 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.

[5] 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.
[6] T.-W. Pan, R. Glowinski, Direct simulation of the motion of neutrally buoyant circular cylinders in plane Poiseuille's flow, J. Comp. Phys. 181 (2002) 260.
[7] C. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220.
[8] N. Patankar, A formulation for fast computations of rigid particulate flows, Center Turbul. Res., Ann. Res. Briefs (2001) 439.
[9] N. Sharma, N. Patankar, A fast computation technique for the direct numerical simulation of rigid particulate flows, J. Comp. Phys. 205 (2005) 439.
[10] J. Guermond, P. Minev, Analysis of a projection/characteristic scheme for incompressible flow, Comm. Numer. Meth. Engng. 19 (2003) 535.
[11] P. Minev, C. Ethier, A semi-implicit projection algorithm for the Navier–Stokes equations with application to flows in complex geometries, Notes on Numerical Fluid Mechanics 73 (1999) 223.
[12] C. Veeramani, P. Minev, K. Nandakumar, A fictitious domain method for particle sedimentation, Lecture Notes in Comp. Sci. 3743 (2005) 544.
[13] A. ten Cate, C. Nieuwstad, J. Derksen, H.V. den Akker, Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity, Phys. Fluids 14 (2002) 4012.
[14] G. Segré, A. Silberberg, Radial particle displacements in Poiseuille's flow of suspensions, Nature 189 (1961) 209.
[15] T.-W. Pan, R. Glowinski, Direct simulation of the motion of neutrally buoyant balls in a three-dimensional Poiseuille's flow, C.R. Mecanique 333 (2005) 884.
[16] B. Yang, J. Wang, D. Joseph, H. Hu, T. Pan, R. Glowinski, Migration of a sphere in a tube flow, J. Fluid Mech. 540 (2005) 109.