A finite element method for particulate flow

E. Bänsch, R. Prignitz

Key words: Particulate flow, Navier-Stokes, finite element method

1 Introduction

Particulate flow, i.e. the flow of a carrier fluid loaded with particles, plays an important role in many technical applications. Let us just mention reactors, fluidized beds, production of nano particles and many more. There exists a hierarchy of models how to describe the particulate phase and how to describe the interaction between particles and fluid. For a comprehensive list of references we refer to the articles of Esmaelli & Tryggvason [6] and Hu [12].

For certain applications it is mandatory to describe the fluid–particle interaction and also a possible particle-particle interaction in full detail without simplified parametrizations. Computational methods based on such full models are called *direct numerical simulations*.

One of the most important points in simulating particulate flow is the numerical representation of the particles' geometry. In Feng et al. and Johnson & Tezduyar [7, 13] a remeshing technique was used to explicitly follow the geometry in time; Wan and Turek [22] introduced a mesh deformation technique and Glowinski et al. [9] used Lagrange multipliers on regular grids. Also immersed boundary methods are very popular, for example LeVeque & Li and Veeramani et al. [14, 20]. Distributed Lagrange multipliers to account for the stress boundary condition are used in Bönisch & Heuveline and Bönisch et al. [5, 4]. In Maury [16] a projection based method was already introduced, still following explicitly the geometry, thus requiring remeshing.

Analytical results regarding existence, uniqueness and qualitative behavior of solutions can be found for instance in Galdi and Serre [8, 19].

The approach presented here is based on the *one domain approach* by [19, 9], but differs from the above mentioned articles in one or several aspects, since it

- does not require an explicit meshing of the particles' domain;
- does not need an explicit evaluation of forces;

E. Bänsch, R. Prignitz
U of Erlangen, Cauerstr. 11, 91058 Erlangen, Germany
e-mail: baensch@math.fau.de
e-mail: prignitz@math.fau.de

- uses a *subspace projection method* to account for the constraint of rigid body motion within the particles, thus avoiding a saddle point problem for this constraint;
- uses time dependent adaptively refined meshes to provide the necessary geometric resolution.

It turns out that this novel method is therefore easy to implement (only few modules have to be added to an existing standard software) and rather efficient. A more detailed presentation can be found in [17].

2 Mathematical Formulation

2.1 Model

In this section we introduce the mathematical model for particulate flows. For ease of presentation we restrict ourselves to the case of a single particle. The extension to more particles is straightforward, simply by adding an index. The model also holds for the $2d$-case, one just has to adapt the definition of the cross-product involved in the equations.

Denote by $\Omega(t) \subset \mathbb{R}^3$ the time-dependent domain occupied by an incompressible, Newtonian fluid with velocity u and pressure p. Its motion is described by the incompressible Navier-Stokes equations. A homogeneous no-slip condition is prescribed on the outer boundary Γ_D.

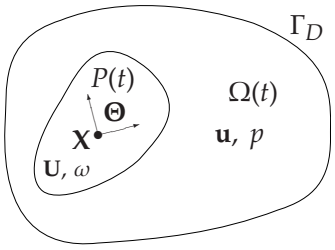

Fig. 1 Particle $P(t)$ of arbitrary shape inside the fluid domain $\Omega(t)$.

$P(t) \subset \mathbb{R}^3$ is the time-dependent domain of a rigid particle, with its center of mass given by $X = \frac{1}{|P(t)|}\int_{P(t)} x\,dx$, while $r = x - X$ is its relative coordinate. The particle's motion, being a rigid body motion, is governed by Newton's law, describing values for the translational and angular velocities U, ω, respectively, and the position X. The orientation in space is given by a complete system of orthogonal unit vectors who's coordinates are denoted by Θ. Since the particle is impermeable, we assume $\Omega(t) \cap P(t) = \emptyset$ for all times $t > 0$. Finally we assume (for simplicity) that the whole

volume $\Omega_c = \Omega(t) \cup P(t) \cup \partial P(t)$ is time independent. See also Fig. 2.1 for a sketch of the situation.

The motions of fluid and particle are coupled on one hand by the *no-slip-condition* on the particle boundary Eq. (4) below and and on the other hand by the stress and pressure forces of the fluid acting on the particle (in the right hand sides of Eq. (5)). The mathematical model consists of a coupled system of partial differential equations (PDE) for u, p and of ordinary differential equations (ODE) for U, ω, X and Θ reading in non-dimensional form

$$\partial_t u + (u \cdot \nabla) u - \nabla \cdot \left(\frac{1}{\mathrm{Re}} \mathrm{D}[u] - p\mathbb{I} \right) = 0 \quad \text{in } \Omega(t), \tag{1}$$

$$\nabla \cdot u = 0 \quad \text{in } \Omega(t), \tag{2}$$

$$u = 0 \quad \text{on } \Gamma_D, \tag{3}$$

$$u = U + \omega \times r \quad \text{on } \partial P(t), \tag{4}$$

$$M\dot{U} = F - \int_{\partial P(t)} \sigma n \, ds, \quad I\dot{\omega} + \omega \times (I\omega) = - \int_{\partial P(t)} r \times \sigma n \, ds, \tag{5}$$

$$\dot{X} = U, \quad \dot{\Theta} = \mathrm{R}[\omega]\Theta. \tag{6}$$

The system has to be closed by appropriate initial conditions. Here, Re is the Reynolds number, M and I the mass and inertia tensors, respectively; $\sigma := \frac{1}{\mathrm{Re}} \mathrm{D}[u] - p\mathbb{I}$ is the stress tensor, where $\mathrm{D}[\cdot]$ is the deformation tensor $\mathrm{D}[u]_{i,j} = \partial_j u_i + \partial_i u_j$. F describes an external force acting on the particle like gravity, particle-particle (in case of more than 1 particle) or particle-wall interaction. $\mathrm{R}[\cdot]$ is the cross-product operator.

2.2 Weak formulation

Following the idea and presentation in [9] a weak formulation of the system Eqs. (1)-(6) is presented. This formulation is instrumental for deriving our numerical method in the next section. Define

$$H_c(\Omega_c) = \left\{ (v, V, \xi) \,\middle|\, v \in \left(H^1(\Omega_c)\right)^3, V \in \mathbb{R}^3, \xi \in \mathbb{R}^3, v = 0 \text{ on } \Gamma_D, v = V + \xi \times r \text{ in } P(t) \right\}. \tag{7}$$

Note that by the above definition the velocity v in $H_c(\Omega_c)$ is defined on the combined domain Ω_c and is restricted to the rigid body velocity $V + \xi \times r$ inside the particle. For a shorter notation we introduce the bi- and trilinear forms

$$m(u, v) = \int_{\Omega_c} u \cdot v \, dx, \tag{8}$$

$$s(u, v) = \frac{1}{2\mathrm{Re}} \int_{\Omega_c} \mathrm{D}[u] : \mathrm{D}[v] \, dx, \tag{9}$$

$$k(w;u,v) = \int_{\Omega_c} (w\cdot\nabla)\, u\cdot v\, dx, \tag{10}$$
$$b(q,v) = \int_{\Omega_c} q\nabla\cdot v\, dx, \tag{11}$$

and the variable $\beta = 1-\alpha$. Then Eqs. (1)–(6) can be compactly written as:

Find $(\mathfrak{u},p)$ with $\mathfrak{u}(t)\in H_c(\Omega_c)$, $p(t)\in L^2_0(\Omega_c)$ such that for all $(\mathfrak{v},q)\in \big(H_c(\Omega_c)\times L^2_0(\Omega_c)\big)$

$$m(\dot u,v)+k(u;v,u)+s(u,v)-b(p,v)+ \beta M\dot U\cdot V+\beta\left(I\dot\omega+\omega\times(I\omega)\right)\cdot\xi = F\cdot V, \tag{12}$$
$$b(q,u) = 0, \tag{13}$$
$$\dot X = U, \tag{14}$$
$$\dot\Theta = \mathrm{R}[\omega]\,\Theta. \tag{15}$$

Eqs. (12) and (13) are called the *combined* Navier-Stokes equations. The time dependence of $\Omega(t)$ and $P(t)$ is now completely coded in the time dependent definition of $H_c(\Omega_c)$.

3 Numerical Method

The numerical scheme to solve the weak problem Eqs. (12)–(15) derived in the previous section consists of the following six points:

1. splitting scheme to decouple the unknowns;
2. a pressure correction projection scheme based on a BDF2 method to efficiently solve the combined Navier-Stokes equations;
3. subspace projection to incorporate the restrictions given by the function space $H_c(\Omega_c)$;
4. adaptivity in space;
5. preconditioning;
6. Barnes-Hut algorithm for particle-particle interaction.

3.1 Splitting by time discretization

Predictor

Given F^k, X^k and U^k.

$$X^{k+1} := X^k + \tau U^k + \frac{\tau^2}{2\beta M} F^k, \quad U^{k+\frac{1}{2}} := U^k + \frac{\tau}{2\beta M} F^k. \tag{16}$$

$$F^{k+1} = F(t^{k+1}, X^{k+1}), \quad \breve{U} := U^{k+\frac{1}{2}} + \frac{\tau}{2\beta M} F^{k+1}. \tag{17}$$

Combined Navier Stokes

Step 1 (Momentum equation)
Given u^k, u^{k-1}, p^k, χ^k, χ^{k-1}, $\breve{U}$, ω^k.
Set $u^\star = 2u^k - u^{k-1}$, $\quad \omega^\star = 2\omega^k - \omega^{k-1}$.

Find $\mathfrak{u}^{k+1} \in H_c(\Omega_c)$ such that for all $\mathfrak{v} \in H_c(\Omega_c)$

$$\begin{aligned} & m(u^{k+1}, v) && +\gamma k(u^\star; u^{k+1}, v) + \gamma s(u^{k+1}, v) + \\ & \frac{2}{3}\beta M U^{k+1} \cdot V && +\tfrac{2}{3}\beta I\omega^{k+1} \cdot \xi + \tfrac{\gamma}{2}\beta\,\omega^\star \times \left(I\omega^{k+1}\right) \cdot \xi = \\ & \gamma b(p^k, v) && +m(\tfrac{4}{3}u^k - \tfrac{1}{3}u^{k-1}, v) + \gamma b(\tfrac{4}{3}\chi^k - \tfrac{1}{3}\chi^{k-1}, v) + \\ & \frac{2}{3}\beta M \breve{U} \cdot V && +\tfrac{2}{3}\beta I\omega^k \cdot \xi - \tfrac{\gamma}{2}\beta\,\omega^k \times \left(I\omega^k\right) \cdot \xi. \end{aligned} \tag{18}$$

Step 2 (Computation of pressure correction)

Find $\chi^{k+1} \in H^1(\Omega_c)$ such that for all $\Psi \in H^1(\Omega_c)$

$$m(\nabla\chi^{k+1}, \nabla\Psi) = \frac{1}{\gamma} b(\Psi, u^{k+1}). \tag{19}$$

Step 3 (Pressure update in rotational form)

Find $p^{k+1} \in L^2_0(\Omega_c)$ such that for all $q \in L^2_0(\Omega_c)$

$$m(p^{k+1}, q) = m(p^k + \chi^{k+1}, q) - b(q, \frac{2}{\mathrm{Re}} u^{k+1}). \tag{20}$$

Corrector

Given Θ^k, X^k, ω^k, ω^{k+1}, U^k and U^{k+1}.

$$\Theta^{k+1} = \left(\mathbb{I} - \frac{\tau}{2}\mathrm{R}\left[\omega^{k+1}\right]\right)^{-1}\left(\mathbb{I} + \frac{\tau}{2}\mathrm{R}\left[\omega^{k}\right]\right)\Theta^k. \tag{21}$$

$$X^{k+1} = X^k + \frac{\tau}{2}\left(U^k + U^{k+1}\right). \tag{22}$$

The above technique is used to to solve the highly coupled, highly nonlinear system of equations. The presented algorithm decouples the position and the orientation of the particles (X and Θ) from the combined Navier-Stokes equations (u, U, ω and p). These are then further decoupled by a pressure correction projection method. Thus the philosophy here is to finally split the complex system into a cascade of simple subproblems rather than using a (maybe more accurate but much more expensive) monolithic approach.

To be more precise, in order to discretize in time, the time interval $(0,T)$ is subdivided by discrete time instants: $0 = t^0 < t^1 < \ldots < t^N = T$. Denote by $\tau_{k+1} := t^{k+1} - t^k$. For simplicity a fixed time step size τ is used: $\tau_k = \tau$ for all $k = 1, \ldots N$. Moreover, define $\gamma = \frac{2}{3}\tau$.

Then in each time step Eqs. (12)–(15) are split into three substeps. The first is a *predictor* step for the new particle position and velocity, X^{k+1}, $\check{U}$, respectively. In the second step values for u^{k+1}, U^{k+1}, ω^{k+1} and p^{k+1} are computed based on a BDF2 scheme. The last step is a *corrector* for X^{k+1}, Θ^{k+1}.

The predictor step is a *Velocity Verlet* method, which is of second order [21, 15] and the common tool in particle dynamics.

The combined Navier Stokes equations are discretized by a projection method in *rotation form*, see [11, 10]. To this end, the time derivative $\partial_t u$ is replaced by a BDF2 scheme having good stability properties, while the equations for $\dot{U}$ and $\dot{\omega}$ are approximated by Crank-Nicolson differences, respectively.

The corrector uses the Crank-Nicolson scheme for time discretization.

3.2 Spatial discretization

The core problem in solving the time discretized system are the combined Navier Stokes equations and in particular Eq. (18). The crucial point in the spatial discretization is to define a discrete counterpart of $H_c(\Omega_c)$ and, moreover, the concrete realization of this non–standard finite element space. A brief description of how to solve this problem is given in the sequel, a more comprehensive presentation can be found in [17].

Let $\mathscr{T}$ be a triangulation of $\overline{\Omega}$. Since we are using the *Taylor–Hood* element for velocity and pressure, the basic finite element space for the velocity is given by the space of piecewise quadratic elements:

$$X(\Omega_c) = \left\{ (v,V,\xi) \,|\, v \in \left(C^0(\overline{\Omega}_c)\right)^2, v \in \left(P^k(T)\right)^2 \forall T \in \mathscr{T}, V \in \mathbb{R}^2, \xi \in \mathbb{R}, v = 0 \text{ on } \Gamma_D \right\}.$$

A discrete subspace of $H_c(\Omega_c)$ is now given by

$$X_c(\Omega_c) = \left\{ (v_c,V,\xi) \in X(\Omega_c) |\ v_c = V + \xi \times r \text{ in } P(t) \right\}.$$

For a given time step k the linear Eq. (18) may be rewritten with the bilinear form a, the corresponding operator $\mathscr{A}$, and the cumulative right hand side g: find $u \in X_c(\Omega_c)$ such that for all $v \in X_c(\Omega_c)$ it holds

$$a(u,v) =: (\mathscr{A}u, v) = (g,v). \tag{23}$$

To circumvent the explicit representation of $H_c(\Omega_c)$, a subspace projection $\pi : X \to X_c$ is used. With this operator (23) may be formulated in terms of the *standard* finite element space $X(\Omega_c)$: find $\tilde{u} \in X(\Omega_c)$ such that for all $v \in X(\Omega_c)$ it holds

$$(\mathscr{A}\pi\tilde{u}, \pi v) = (g, \pi v). \tag{24}$$

Note that the solution u is now easily found by taking $u = \pi\tilde{u}$, where $\tilde{u}$ is a solution of Eq. (24). The above system now leads to the linear system of equations for the nodal vector $\tilde{U}$ of the form

$$\Pi^T A \Pi \tilde{U} = \Pi^T G, \tag{25}$$

where A is the system matrix corresponding to operator $\mathscr{A}$ and Π is a matrix representation of π. We call this method *subspace projection method*. Note that, when using iterative solvers, one can bypass to explicitly compute the modified system matrix $\Pi^T A \Pi$, but rather just needs to slightly modify the matrix vector product, because one only has to take into account the action of $\Pi^T A \Pi$ on a vector. Because the matrix Π is quite simple, its not necessary to store it explicitly. Instead, a short routine can perform the multiplication of Π and Π^T with a vector v. The following pseudo–code shows this computation.

```
! Multiplication (u,U,omega)=Pi*(v,V,xi)
subroutine Pmul(v,V,xi,u,U,omega)

   ! U, omega
   do ii=1,npart ! Number of particles
      U(:,ii)    = V(:,ii)
      omega(ii)  = xi(ii)
   end do

   ! u = rigid body motion in the particle
   do i=1,nk ! Number of DOFs
      if( isparticle(i) ) then
         ii= numpart(i)
```

```
         r(:)= x(:,i) - xpart(:,ii)
         u(1,i) = V(1,ii) - r(2)*xi(ii)
         u(2,i) = V(2,ii) + r(1)*xi(ii)
      else
         u(:,i)= v(:,i)
      end if
   end do

end subroutine
```

3.3 Adaptivity

One of the most important issues in simulating particulate flow is the numerical representation of the particle's geometry.

In Hu [12] a remeshing technique was used to explicitly follow the geometry in time, Wan and Turek [22] introduced a mesh deformation technique and Glowinski et al. [9] used Lagrange multipliers. In contrast to these methods, we use time dependent adaptively refined/coarsened grids based on the bisection method [1] to sufficiently resolve the region around the particle.

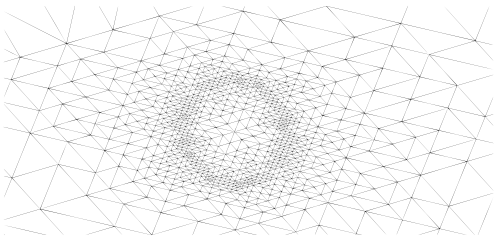

Fig. 2 Adaptive refined mesh around a particle. For an accurate representation it is useful to refine the mesh on the particle boundary.

The overall algorithm was implemented in the finite element flow solver NAVIER, for more details see [2].

3.4 Preconditioning

In general, the matrix $\Pi^T A \Pi$ (if the kernel would be factored out) is ill conditioned so that preconditioning is mandatory for an efficient solution strategy.

We developed a preconditioner based on inexact factorization that gave rather satisfying results, see [18].

3.5 Particle interaction

Efficient evaluation of the particle-particle interactions in Eq. (17) is crucial. For a large number of particles (more than, say, 1000) a naive implementation requiring $\mathcal{O}(n^2)$, n the number of dofs, would be prohibitive. Instead we use the Barnes-Hut algorithm, which reduces the complexity to $\mathcal{O}(n\log(n))$ with an acceptable loss of accuracy, see [3].

The idea of the algorithm is to merge the forces created by a group of neighboring particles into a single force of one pseudo-particle.

In addition to the long range Coulomb forces we also add short range repulsive forces in order to model particle collisions and avoid mutual penetration of particles. A similar approach is used for near particle-wall collision.

4 Computational examples

In this section we present some applications of the method described above. Quantitative validations can be found in [17]. Here we present further numerical experiments.

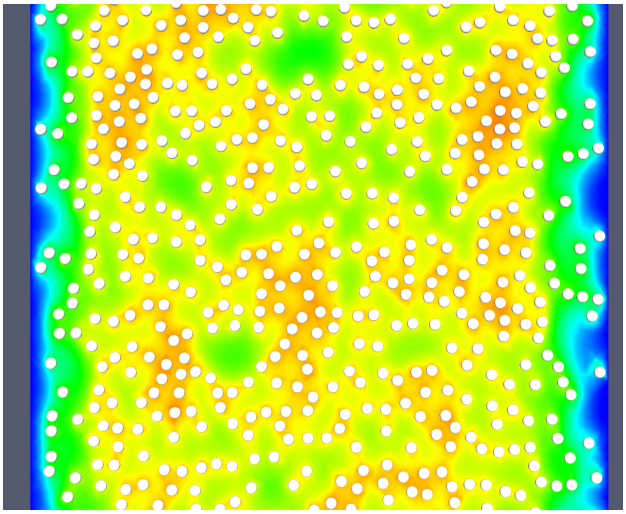

Fig. 3 Sedimentation of particles in 2D.

Fig. 3 shows a snapshot of a bunch of sedimenting particles (in 2D) under the influence of gravity.

The next experiment considers the sedimentation of two spherical particles in a cylindrical domaini in 3D. The particles are initially aligned on the center line, sep-

arated by a small distance of a few particle diameters in the starting configuration. When gravity starts acting one can observe the following situations.

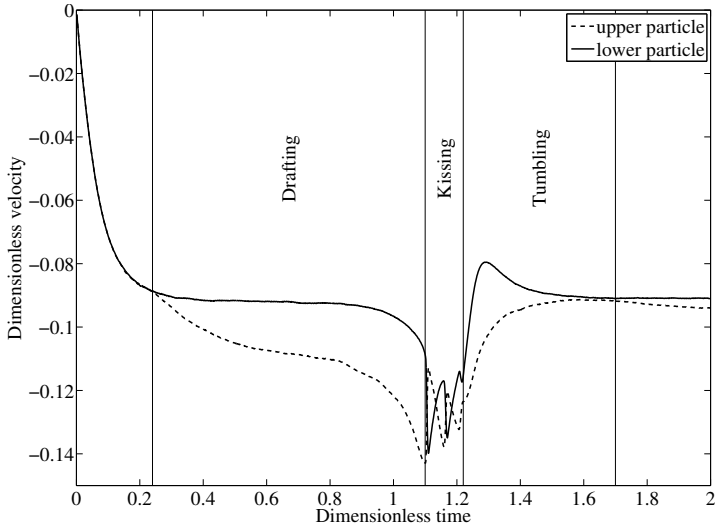

Fig. 4 Velocities of two particles traversing the phases of drafting – kissing – tumbling.

- Both particles start accelerating. There is no interaction between them.
- "Drafting": after a while the slipstream of the first particle causes the second one to accelerate a little more.
- "Kissing": a near impact is inevitable as the second particle has a higher velocity than the first one. The slower particle is pushed by the faster one (the force is transferred by the viscous fluid).
- "Tumbling": the above situation is unstable. To solve this conflict the slower particle moves aside, so that the faster particle can pass it. This can be interpreted as tumbling, when observed in relative coordinates.

These four phases described are displayed in Fig. 4.

The last example is a snapshot of the lid driven cavity in 3D with 1000 immersed particles, Fig. 5.

5 Discussion and Conclusion

A novel finite element method for the simulation of particulate flows was presented. Its key ingredients are: one domain approach, splitting in time, subspace projection method to account for the rigid body motion within the particles and time dependent

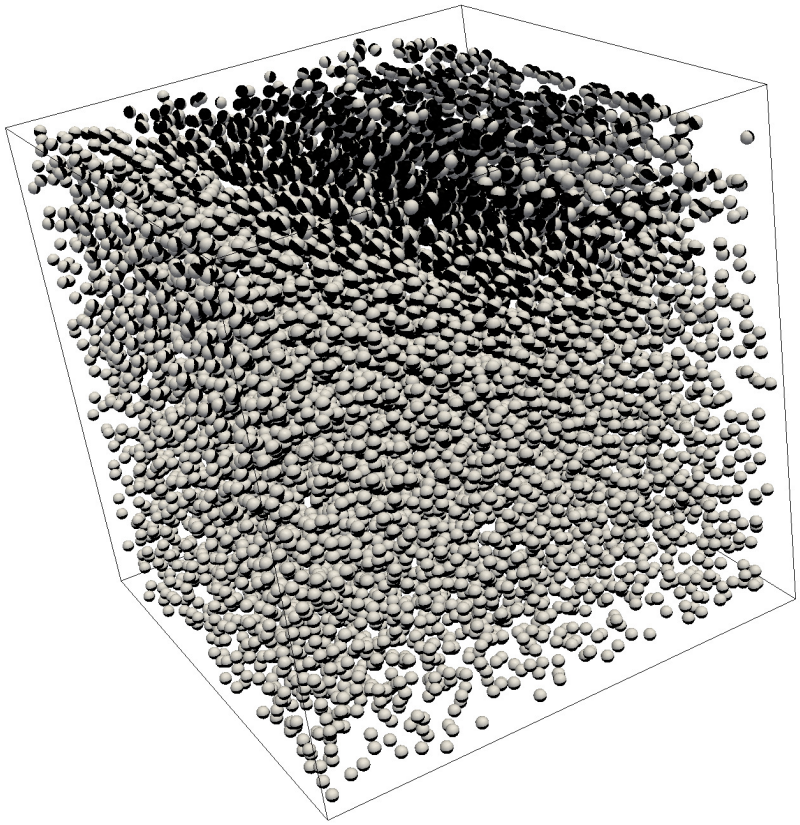

Fig. 5 Lid driven cavity 3D with 1000 particles.

adaptively refined meshes. The advantages of the method are its easy implementation and its efficiency. Only few modifications are needed to extend an existing Navier-Stokes code to simulate particulate flows by this method.

Acknowledgments

This work was supported by the *Bayerische Forschungsstiftung* Grant 813-08, which is gratefully acknowledged.

References

1. Bänsch, E.: Local mesh refinement in 2 and 3 dimensions. Impact Comput. Sci. Engrg. **3**, 181–191 (1991)
2. Bänsch, E.: Simulation of instationary, incompressible flows. Acta Math. Univ. Comenianae **67**(1), 101–114 (1998)
3. Barnes, J., Hut, P.: A hierarchical $\mathscr{O}(n \log n)$ force-calculation algorithm. Nature **324**, 446–449 (1986)
4. Bönisch, S., Dunne, T., Rannacher, R.: Numerics of fluid-structure interaction. In: Hemodynamical flows, *Oberwolfach Semin.*, vol. 37, pp. 333–378. Birkhäuser, Basel (2008)
5. Bönisch, S., Heuveline, V.: On the numerical simulation of the instationary free fall of a solid in a fluid. I. The Newtonian case. Comput. Fluids **36(9)**, 1434–1445 (2007)
6. Esmaeeli, A., Tryggvason, G.: Direct numerical simulations of bubbly flows. Part 1: low Reynolds number arrays. J. Fluid Mech. **377**, 313–345 (1998)
7. Feng, J., Hu, H., Joseph, D.: Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 1. Sedimentation. J. Fluid Mech. **261**, 95–134 (1994)
8. Galdi, G.: Chapter 7: On the motion of a rigid body in a viscous liquid: A mathematical analysis with applications. pp. 653–791. North-Holland (2002)
9. Glowinski, R., Pan, T.W., Hesla, T., Joseph, D.: A distributed Lagrange multiplier/fictitious domain method for particulate flows. Int. J. Multiphase Flow **25**, 755–794 (1999)
10. Guermond, J.L., Minev, P., Shen, J.: An overview of projection methods for incompressible flows. Comput. Meth. Appl. Mech. Eng. **195**, 6011–6045 (2006)
11. Guermond, J.L., Shen, J.: On the error estimates for the rotational pressure-correction projection methods. Math. Comp. **73**, 1719–1737 (2004)
12. Hu, H.: Direct simulation of flows of solid-liquid mixtures. Int. J. Multiphase Flow **22**(2), 335–352 (1996)
13. Johnson, A., Tezduyar, T.: Simulation of multiple spheres falling in a liquid-filled tube. Comput. Methods Appl. Mech. Engrg. **134**, 351–373 (1996)
14. LeVeque, R., Li, Z.: Immersed interface methods for Stokes flow with elastic boundaries or surface tension. SIAM J. Sci. Comput. **18**, 709–735 (1997)
15. Martys, N., Mountain, R.: Velocity Verlet algorithm for dissipative-particle-dynamics-based models of suspensions. Phys. Rev. E **59**(3), 3733–3736 (1999)
16. Maury, B.: Direct simulation of 2d fluid-particle flows in biperiodic domains. J. Comput. Phys. **156**, 325–351 (1999)
17. Prignitz, R., Bänsch, E.: Simulation of particulate electrodynamc flows with the subspace projection method. Submitted
18. Reitsam, A.: Hierarchical preconditioning for P^TAP-systems. Diploma, FAU Erlangen-Nrnberg (2011)
19. Serre, D.: Chute libre dun solide dans un fluide visqueux incompressible. Existence. Jpn. J. Ind. Appl. Math. **4**(1), 99–110 (1987)
20. Veeramani, C., Minev, P., Nandakumar, K.: A fictitious domain formulation for flows with rigid particles: A non-Lagrange multiplier version. J. Comput. Phys. **224**(2), 867–879 (2007)
21. Verlet, L.: Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Phys. Rev. **159**, 98–103 (1967)
22. Wan, D., Turek, S.: Fictitious boundary and moving mesh methods for the numerical simulation of rigid particulate flows. J. Comput. Phys. **222**, 28–56 (2007)