Journal of Computational Physics 231 (2012) 3246–3263

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Efficient symmetric positive definite second-order accurate monolithic solver for fluid/solid interactions

Frédéric Gibou [a,b], Chohong Min [c,*]

[a] Mechanical Engineering Department, University of California, Santa Barbara, CA 93106, United States
[b] Computer Science Department, University of California, Santa Barbara, CA 93106, United States
[c] Mathematics Department, Ewha Womans University, Seoul 120-750, South Korea

ARTICLE INFO

Article history:
Received 9 September 2010
Received in revised form 22 June 2011
Accepted 3 January 2012
Available online 20 January 2012

Keywords:
Fluid
Solid
Rigid body
Interaction between fluid and solid
Level-set method
Navier–Stokes equations

ABSTRACT

We introduce a robust and efficient method to simulate strongly coupled (monolithic) fluid/rigid-body interactions. We take a fractional step approach, where the intermediate state variables of the fluid and of the solid are solved independently, before their interactions are enforced via a projection step. The projection step produces a symmetric positive definite linear system that can be efficiently solved using the preconditioned conjugate gradient method. In particular, we show how one can use the standard preconditioner used in standard fluid simulations to precondition the linear system associated with the projection step of our fluid/solid algorithm. Overall, the computational time to solve the projection step of our fluid/solid algorithm is similar to the time needed to solve the standard fluid-only projection step. The monolithic treatment results in a stable projection step, i.e. the kinetic energy does not increase in the projection step. Numerical results indicate that the method is second-order accurate in the L^∞-norm and demonstrate that its solutions agree quantitatively with experimental results.

1. Introduction

Understanding and predicting fluid–structure interaction is crucial in many areas of science and engineering. Examples include the study of particle motion in liquids, with application to industrial solidification or pertaining to the understanding of sedimentation in geology, the study of heart valves or aneurisms, or the design of engineering systems such as swimming structures. Considerable work has been done on the design of numerical methods to provide an accurate predictive tool for fluid/rigid-body coupling. Arbitrary Lagrangian–Eulerian (ALE) schemes have been successfully employed in the case where the structure deformation is low [10]. Schemes based on the Lattice–Bolztmann method have also been used (see e.g. the recent work of [12]). Immersed boundary methods have provided a framework for coupling fluids and rigid or elastic bodies. Several applications of such methods exist in the literature, with maybe the most famous one being the application of blood flow in the heart [25,26]. Within this framework, the coupling is expressed through the use of a delta formulation, which smears some of the variables near the interface but provides a straightforward approach. Coquerelle and Cottet [8] introduced a vortex method [9] for the simulation of the interaction of an incompressible flow with rigid bodies. In particular, they consider a single flow and use a penalization technique to enforce continuity at the solid–fluid interface and rigid motion inside the solid. In this case, the interface is represented by a level-set function and the quantities are smeared across the interface.

* Corresponding author.
E-mail addresses: fgibou@enginnering.ucsb.edu (F. Gibou), chohong@ewha.ac.kr (C. Min).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2012.01.009

In this paper, we are focusing on the two-way coupling between a fluid and a rigid body and present a sharp treatment for their coupling. In this context, existing numerical methods fall into two categories: partitioned approaches, where equations for the fluid are first solved before equations for the solid are solved, and monolithic approaches, where equations for the fluid and the rigid body are solved simultaneously [1,11]. An advantage of the partitioned approach is that existing fluid and structure solvers can be used in an iterative process. A disadvantage is that it is difficult to develop methods that guarantee numerical stability, which in practice can translate into spurious numerical oscillations. Also, the accuracy may suffer from the lack of strong coupling, even if this can somewhat be remedied with sub-iteration procedures. Monolithic approaches have the potential of being more stable and more accurate, but their design and analysis is not as straightforward.

In [3], Batty et al. have formulated the interactions between fluids and non-deformable solids as a kinetic energy minimization problem, and discretized the interactions through the corresponding Euler–Lagrange equation. In [28], Robinson-Mosher et al. have presented a monolithic approach based on a projection framework. A particularly important feature of this approach is the design of a symmetric positive definite (SPD) system for the projection step, an improvement on their previous work that produced an indefinite system [29]. Designing a SPD system guarantees that its solution can be computed with fast solvers. Their method also preserves momentum, which impacts positively on the stability of the method. However the method is only first-order accurate in the L^∞-norm and it is not clear how to best precondition the linear system.

In the present paper, we present an SPD approach that is second-order accurate in the L^∞-norm and for which the linear system can be preconditioned easily. Our method is unconditionally stable and conserves the momentum transfer between the fluid and the solid in the projection step. We take a fractional step approach, where the intermediate state variables of the fluid and of the solid are solved independently, before their interactions are enforced via a projection step. Using the Heaviside function of the fluid region, the projection is formulated as a simple Poisson-type equation and can be easily implemented as a small addition to the standard projection method for fluids on a MAC grid. We show how one can use the standard preconditioner used in fluid simulations to efficiently precondition the linear system associated with our fluid/solid coupling projection step, providing a simulation framework as efficient as standard fluid solvers. We present numerical results in two and three spatial dimensions that indicate that our method is second-order accurate. We also provide simulation results that are in agreement with the experimental results of Cate et al. [6].

2. Governing equations

The interactions between a fluid and a rigid body are modeled by equations of motion for the fluid and the solid, as well as their respective boundary conditions. We consider a computational domain $\Omega = \Omega^f \cup \Omega^s$ with boundary $\partial\Omega$, where a rigid body defined by Ω^s with boundary Γ is immersed in a fluid defined in a region $\Omega^f \subset \Omega$; see the schematic in Fig. 2.1.

2.1. Fluid equations

In the case where the viscosity is constant the motion of incompressible flows is described by the incompressible Navier–Stokes equations of the following forms:

$$\begin{aligned} &\boldsymbol{U}_t + (\boldsymbol{U}\cdot\nabla)\boldsymbol{U} + \frac{\nabla p}{\rho} = \frac{\mu}{\rho}\Delta\boldsymbol{U} + \boldsymbol{g} \text{ in } \Omega^f,\\ &\nabla\cdot\boldsymbol{U} = 0, \text{ in } \Omega^f, \end{aligned}$$

where ρ is the fluid's density, μ is the fluid's viscosity, $\boldsymbol{U} = (u, v, w)$ is the velocity field and $\boldsymbol{g}$ the gravity field.

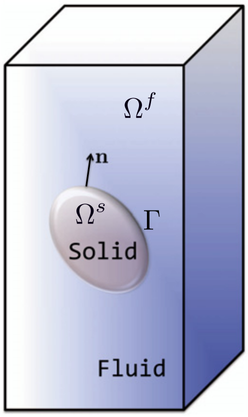


Fig. 2.1. Schematic of a typical fluid/solid computational set up.

2.2. Solid equations

The solid motion is fully described by the evolution of its center of mass $\boldsymbol{C}(t)$, its linear momentum $\boldsymbol{P}(t)$, its angular momentum $\boldsymbol{L}(t)$, and the orientation matrix $\underline{\underline{\boldsymbol{R}}}(t)$:

$$\begin{aligned}
&\frac{d\boldsymbol{C}}{dt} = \frac{\boldsymbol{P}}{m},\\
&\frac{d\boldsymbol{P}}{dt} = m\boldsymbol{g} + \boldsymbol{f},\\
&\frac{d\boldsymbol{L}}{dt} = \boldsymbol{\tau},\\
&\frac{d\underline{\underline{\boldsymbol{R}}}}{dt} = \underline{\underline{\boldsymbol{\omega}}}^{\times}\underline{\underline{\boldsymbol{R}}},
\end{aligned}$$

where m is the mass of the rigid body, $\boldsymbol{\omega} = \underline{\underline{I}}^{-1}\boldsymbol{L}$ is the angular velocity defining the inertia matrix $\underline{\underline{I}}$, and $\underline{\underline{\boldsymbol{\omega}}}^{\times}$ is the skew-symmetric matrix corresponding to the cross product of any vector $\boldsymbol{x}$ with ω, i.e. $\underline{\underline{\boldsymbol{\omega}}}^{\times}\boldsymbol{x} = \boldsymbol{\omega} \times \boldsymbol{x}$ for all $\boldsymbol{x}$. The force and the torque exerted by the fluid are denoted by $\boldsymbol{f}$ and $\boldsymbol{\tau}$, respectively.

2.3. Boundary conditions

The fluid/solid two-way coupling is described by the boundary conditions: the fluid must satisfy the no-slip and non-penetrating boundary conditions imposed by the solid, which takes into account the rigid body's linear velocity $\boldsymbol{v} = \frac{1}{m}\boldsymbol{P}$ and the rigid body's angular velocity, while the solid must be correctly accelerated by the force and torque induced by the fluid stress tensor $\underline{\underline{\boldsymbol{\sigma}}} = -p\underline{\underline{\boldsymbol{I}}} + 2\mu\underline{\underline{\boldsymbol{D}}}$, with $\underline{\underline{\boldsymbol{I}}}$ the identity matrix and $\underline{\underline{\boldsymbol{D}}}$ the standard strain rate tensor:

$$\begin{aligned}
&\boldsymbol{U}(x,t) = \boldsymbol{U}^{\text{solid}}(\boldsymbol{x},t) := \frac{\boldsymbol{P}(t)}{m} + \boldsymbol{\omega}(t) \times (\boldsymbol{x} - \boldsymbol{C}(t)) \text{ on } \Gamma,\\
&\boldsymbol{f}(t) = \int_{\Gamma} (-p\underline{\underline{\boldsymbol{I}}} + 2\mu\underline{\underline{\boldsymbol{D}}}) \cdot \boldsymbol{n}\, d\Gamma,\\
&\boldsymbol{\tau}(t) = \int_{\Gamma} (\boldsymbol{x} - \boldsymbol{C}) \times (-p\boldsymbol{I} + 2\mu\underline{\underline{\boldsymbol{D}}}) \cdot \boldsymbol{n}\, d\Gamma,
\end{aligned}$$

where $\boldsymbol{n}$ is the outward normal to the rigid body.

3. Temporal discretization

We use a projection method, where the evolution of the different state variables is performed in two stages: intermediate states are first computed (see Sections 3.1 and 3.2) before being projected so that the divergence free condition for the velocity field $\boldsymbol{U}^{n+1}$ at t^{n+1} is satisfied as well as to enforce the boundary conditions for $\boldsymbol{U}^{n+1}$ (see Sections 3.3 and 3.4). In particular, we use a projection method with pressure guess, i.e. we use the pressure p^n at time t^n when computing the intermediate states.

3.1. Intermediate states for the fluid equations

In the case of incompressible flows, shocks are not present so that a standard semi-Lagrangian method is appropriate for discretizing the convection term of the Navier–Stokes equations. A benefit of this approach is that it produces second-order accurate discretizations that are unconditionally stable [36,20]. The diffusion term $\mu\,\Delta\boldsymbol{U}$ is traditionally discretized with the Crank–Nicolson scheme. However, within the standard projection method, the combination of the Crank–Nicolson scheme to discretize the diffusion term and the semi-Lagrangian method to discretize the momentum term produces solutions that are only first-order accurate [36]. We therefore prefer to discretize the diffusion term with the second-order backward difference formula (BDF), which leads to second-order accurate solutions in conjunction with the semi-Lagrangian method [36,20]: denoting $\boldsymbol{U}^{*}$ the intermediate velocity field sampled at a grid node $\boldsymbol{x}^{n+1}$, its evolution in time is given by:

$$\begin{aligned}
&\rho\frac{\frac{3}{2}\boldsymbol{U}^{*} - 2\boldsymbol{U}_d^{n} + \frac{1}{2}\boldsymbol{U}_d^{n-1}}{\Delta t} + \nabla p_d^{n} = \mu\Delta\boldsymbol{U}^{*} + \rho\boldsymbol{g} \text{ in } \Omega^{\mathrm{f},n+1},\\
&\boldsymbol{U}^{*} = \boldsymbol{U}^{\text{solid},n+1} \text{ on } \Gamma^{n+1},
\end{aligned} \tag{3.1}$$

where Δt is the time step. The Lagrangian coordinates $\boldsymbol{x}(t)$ are convected back in time along the characteristic curves according to the ordinary differential equation $\boldsymbol{x}'(t) = \boldsymbol{U}(x,t)$ with initial data $\boldsymbol{x}(t^{n+1}) = \boldsymbol{x}^{n+1}$. This defines a departure point $\boldsymbol{x}_d = \boldsymbol{x}(t^n)$ for every grid node $\boldsymbol{x}^{n+1}$, at which the velocity fields $\boldsymbol{U}_d^{n}$ at time t^n, the velocity field $\boldsymbol{U}_d^{n-1}$ at time t^{n-1} and the pressure field p_d^{n} at time t^n are computed by the stabilized quadratic interpolation in [21].

3.2. Intermediate states for the solid equations

The solid equations are evolved in such a way as to mimic the time evolution of the fluid's states, i.e. we use the same BDF scheme. In particular, we compute intermediate states by explicitly isolating the contribution of the pressure term in order to have a unifying framework for the projection step of Section 3.4:

$$\begin{aligned}
&\frac{\frac{3}{2}\boldsymbol{C}^{n+1}-2\boldsymbol{C}^{n}+\frac{1}{2}\boldsymbol{C}^{n-1}}{\Delta t}=2\frac{\boldsymbol{P}^{n}}{m}-\frac{\boldsymbol{P}^{n-1}}{m},\\
&\frac{\frac{3}{2}\boldsymbol{P}^{*}-2\boldsymbol{P}^{n}+\frac{1}{2}\boldsymbol{P}^{n-1}}{\Delta t}=2\tilde{\boldsymbol{f}}^{n}-\tilde{\boldsymbol{f}}^{n-1}+\int_{\Gamma^{n+1}}p^{n}\boldsymbol{n}\,d\Gamma^{n+1}+m\boldsymbol{g},\\
&\frac{\frac{3}{2}\boldsymbol{L}^{*}-2\boldsymbol{L}^{n}+\frac{1}{2}\boldsymbol{L}^{n-1}}{\Delta t}=2\tilde{\boldsymbol{\tau}}^{n}-\tilde{\boldsymbol{\tau}}^{n-1}+\int_{\Gamma^{n+1}}(\boldsymbol{x}-\boldsymbol{C}^{n+1})\times p^{n}\boldsymbol{n}\,d\Gamma^{n+1},\\
&\frac{\frac{3}{2}\underline{\underline{\boldsymbol{R}}}^{n+1}-2\underline{\underline{\boldsymbol{R}}}^{n}+\frac{1}{2}\underline{\underline{\boldsymbol{R}}}^{n-1}}{\Delta t}=2(\underline{\underline{\boldsymbol{\omega}}}^{\times}\underline{\underline{\boldsymbol{R}}})^{n}-(\underline{\underline{\boldsymbol{\omega}}}^{\times}\underline{\underline{\boldsymbol{R}}})^{n-1},
\end{aligned}\tag{3.2}$$

where $\tilde{\boldsymbol{f}}=\int(2\mu\underline{\underline{\boldsymbol{D}}})\boldsymbol{n}d\Gamma$ and $\tilde{\boldsymbol{\tau}}=\int(\boldsymbol{x}-\boldsymbol{C})\times(2\mu\underline{\underline{\boldsymbol{D}}})\boldsymbol{n}d\Gamma$, i.e. the force and torque without their pressure component. We have also approximated $\boldsymbol{P}^{n+1}$, $\tilde{\boldsymbol{f}}^{n+1}$, $\tilde{\boldsymbol{\tau}}^{n+1}$ and $(\underline{\underline{\boldsymbol{\omega}}}^{\times}\boldsymbol{R})^{n+1}$ by linear extrapolation in time, e.g. $\boldsymbol{P}^{n+1}\approx 2\boldsymbol{P}^{n}-\boldsymbol{P}^{n-1}$.

3.3. Boundary conditions and Heaviside formulation

The boundary conditions are given on the interface Γ and since it is more convenient to express them on the entire domain Ω, we consider their influence through the use of a Heaviside formulation. Specifically, let $H(\boldsymbol{x},t)$ be the Heaviside function equal to 1 in the fluid region and 0 in the solid region. Since $\nabla H=\delta_{\Gamma}\boldsymbol{n}$, where δ is the Dirac delta function with support on Γ and $\boldsymbol{n}$ is the outward normal vector at Γ, we can rewrite the boundary conditions as:

$$\begin{aligned}
\boldsymbol{f}&=\int_{\Gamma}(-p\underline{\underline{\boldsymbol{I}}}+2\mu\underline{\underline{\boldsymbol{D}}})\boldsymbol{n}\,d\Gamma \iff \boldsymbol{f}=\int_{\Omega}(-p\underline{\underline{\boldsymbol{I}}}+2\mu\underline{\underline{\boldsymbol{D}}})\nabla H\,d\Omega,\\
\boldsymbol{\tau}&=\int_{\Gamma}(\boldsymbol{x}-\boldsymbol{C})\times(-p\underline{\underline{\boldsymbol{I}}}+2\mu\underline{\underline{\boldsymbol{D}}})\boldsymbol{n}\,d\Gamma \iff \boldsymbol{\tau}=\int_{\Omega}(\boldsymbol{x}-\boldsymbol{C})\times(-p\underline{\underline{\boldsymbol{I}}}+2\mu\underline{\underline{\boldsymbol{D}}})\nabla H\,d\Omega.
\end{aligned}$$

Likewise, the scalar variable q, described in Section 3.4 and satisfying the boundary condition $\nabla q\cdot\boldsymbol{n}=0$ on Γ can be expressed as:

$$\nabla q\cdot\nabla H=0 \text{ on } \Omega.$$

3.4. Projection step

The intermediate variables $\boldsymbol{U}^{*}$, $\boldsymbol{P}^{*}$ and $\boldsymbol{L}^{*}$ must be projected to satisfy the incompressibility condition $\nabla\cdot\boldsymbol{U}^{n+1}=0$ as well as the non-penetration condition $\boldsymbol{U}^{n+1}\cdot\boldsymbol{n}=\boldsymbol{U}^{\text{solid},n+1}\cdot\boldsymbol{n}$ on Γ. The Hodge theorem allows to decompose the intermediate velocity field $\boldsymbol{U}^{*}$ as $\boldsymbol{U}^{*}=\boldsymbol{U}^{n+1}+\Delta t\frac{\nabla q^{n+1}}{\rho^{n+1}}$, with $\boldsymbol{U}^{n+1}$ a divergence free velocity field and q^{n+1} a scalar function. This is the basis of the original projection method of Chorin [7], which has been extended to higher order accuracy and analyzed by other researchers (see e.g. [18,5,4] and the references therein). Similarly, we choose to decompose the intermediate linear momentum $\boldsymbol{P}^{*}$ and angular momentum $\boldsymbol{L}^{*}$ in their corresponding form so that the projection step can be uniformly written as:

$$\boldsymbol{U}^{n+1}=\boldsymbol{U}^{*}-\Delta t\frac{\nabla q^{n+1}}{\rho^{n+1}},\tag{3.3a}$$

$$\boldsymbol{P}^{n+1}=\boldsymbol{P}^{*}-\Delta t\int q^{n+1}\cdot\nabla H^{n+1},\tag{3.3b}$$

$$\boldsymbol{L}^{n+1}=\boldsymbol{L}^{*}-\Delta t\int(\boldsymbol{x}-\boldsymbol{C}^{n+1})\times q^{n+1}\cdot\nabla H^{n+1}.\tag{3.3c}$$

Here H^{n+1} denotes the Heaviside function of the fluid's domain $\Omega^{f,n+1}$. The incompressibility and non-penetration conditions provide an equation for q^{n+1}: we have from the incompressibility condition $H^{n+1}\nabla\cdot\boldsymbol{U}^{n+1}=0$ in the entire domain and since the solid is incompressible, we have $\nabla\cdot\boldsymbol{U}^{\text{solid},n+1}=0$. From the non-penetration boundary condition $\boldsymbol{U}^{n+1}\cdot\boldsymbol{n}=\boldsymbol{U}^{\text{solid},n+1}\cdot\boldsymbol{n}$ on Γ^{n+1}, we have $(\boldsymbol{U}^{n+1}-\boldsymbol{U}^{\text{solid},n+1})\cdot\nabla H^{n+1}=0$. Therefore:

$$\nabla\cdot(H^{n+1}(\boldsymbol{U}^{n+1}-\boldsymbol{U}^{\text{ solid},n+1}))=H^{n+1}\left(\nabla\cdot\boldsymbol{U}^{n+1}+\underbrace{\nabla\cdot\boldsymbol{U}^{\text{solid},n+1}}_{0}\right)+\underbrace{\nabla H^{n+1}\cdot\left(\boldsymbol{U}^{n+1}-\boldsymbol{U}^{\text{solid},n+1}\right)}_{0}=\underbrace{H^{n+1}\nabla\cdot\boldsymbol{U}^{n+1}}_{0}=0.$$

Now combining this expression with equations in (3.3), we have:

$$
\begin{aligned}
0 &= \nabla \cdot \left(H^{n+1} \left(\boldsymbol{U}^* - \Delta t \cdot \frac{\nabla q^{n+1}}{\rho^{n+1}} \right) \right) - \nabla H^{n+1} \cdot (\boldsymbol{v}^{n+1} + \boldsymbol{\omega}^{n+1} \times (\boldsymbol{x} - \boldsymbol{C}^{n+1})) \\
&= \nabla \cdot \left(H^{n+1} \left(\boldsymbol{U}^* - \Delta t \cdot \frac{\nabla q^{n+1}}{\rho^{n+1}} \right) \right) - \nabla H^{n+1} \cdot \frac{1}{m} \left(\boldsymbol{P}^* - \Delta t \cdot \int q^{n+1} \cdot \nabla H^{n+1}\right) - \nabla H^{n+1} \\
&\quad \cdot \left(\left((\underline{\underline{I}}^{-1})^{n+1} \left(\boldsymbol{L}^* - \Delta t \cdot \int (\boldsymbol{x} - \boldsymbol{C}^{n+1}) \times q^{n+1} \cdot \nabla H^{n+1} \right) \right) \times (\boldsymbol{x} - \boldsymbol{C}^{n+1}) \right).
\end{aligned}
$$

Finally, collecting all the q^{n+1} terms on the left-hand-side, we obtain the following linear system for q^{n+1}:

$$
\begin{aligned}
&-\nabla \cdot \left(\frac{H^{n+1}}{\rho} \nabla q^{n+1} \right) + \nabla H^{n+1} \cdot \frac{1}{m} \left(\int q^{n+1} \nabla H^{n+1} \right) + \boldsymbol{J}^{n+1} \cdot \left((\underline{\underline{I}}^{-1})^{n+1} \left(\int q^{n+1} \boldsymbol{J}^{n+1} \right) \right) \\
&\quad = -\frac{1}{\Delta t} \nabla \cdot (H \boldsymbol{U}^*) + \frac{1}{\Delta t} \nabla H \cdot \boldsymbol{U}^{\text{solid},*},
\end{aligned}
\tag{3.4}
$$

where $\boldsymbol{J}^{n+1} = (\boldsymbol{x} - \boldsymbol{C}^{n+1}) \times \nabla H^{n+1}$ and $\boldsymbol{U}^{\text{solid},*} = \frac{\boldsymbol{P}^*}{m} + \left(\left(\underline{\underline{I}}^{-1} \right)^{n+1} \boldsymbol{L}^* \right) \times (\boldsymbol{x} - \boldsymbol{C}^{n+1})$. The following theorem demonstrates that the linear system associated with the above temporal discretization is symmetric and positive definite. The spatial approximations described in Section 4 will be designed to keep this property.

Theorem 1. *The linear system associated with the semi-discrete equation (3.4) is symmetric and positive definite.*

Proof. Let $\mathfrak{L}(\cdot)$ be the operator in the left-hand-side of (3.4) acting on q^{n+1}. To prove the symmetry of the operator $\mathfrak{L}$, we show that $\int_{\mathbb{R}^d} \mathfrak{L}(q^{(1)}) q^{(2)} = \int_{\mathbb{R}^d} \mathfrak{L}(q^{(2)}) q^{(1)}$ for any L^2-integrable functions $q^{(1)}$ and $q^{(2)}$:

$$
\begin{aligned}
\int_{\mathbb{R}^d} \mathfrak{L}(q^{(1)}) q^{(2)} &= -\int_{\mathbb{R}^d} \left(\nabla \cdot \left(\frac{H}{\rho} \nabla q^{(1)} \right) \right) q^{(2)} + \int_{\mathbb{R}^d} \nabla H \cdot \left(\int_{\mathbb{R}^d} q^{(1)} \frac{1}{m} \nabla H \right) q^{(2)} + \int_{\mathbb{R}^d} \underline{\underline{I}}^{-1} \boldsymbol{J} \cdot \left(\int_{\mathbb{R}^d} q^{(1)} \boldsymbol{J} \right) q^{(2)} \\
&= \int_{\mathbb{R}^d} \frac{H}{\rho} \nabla q^{(1)} \cdot \nabla q^{(2)} + \frac{1}{m} \left(\int_{\mathbb{R}^d} q^{(2)} \nabla H \right) \cdot \left(\int_{\mathbb{R}^d} q^{(1)} \nabla H \right) + \left(\underline{\underline{I}}^{-1} \left(\int_{\mathbb{R}^d} q^{(2)} \boldsymbol{J} \right) \right) \cdot \left(\int_{\mathbb{R}^d} q^{(1)} \boldsymbol{J} \right).
\end{aligned}
$$

This is clearly a symmetric form for $q^{(1)}$ and $q^{(2)}$, since the inertia matrix and its inverse are symmetric. The positive definiteness of the operator $\mathfrak{L}$ is a straightforward consequence of the fact that the inertia matrix is positive definite: For any L^2-integrable function q, we have:

$$
\int_{\mathbb{R}^d} \mathfrak{L}(q) q = \int_{\mathbb{R}^d} \frac{H}{\rho} \|\nabla q\|^2 + \frac{1}{m} \left\| \int_{\mathbb{R}^d} q \nabla H \right\|^2 + \left(\underline{\underline{I}}^{-1} \left(\int_{\mathbb{R}^d} q \boldsymbol{J} \right) \right) \cdot \left(\int_{\mathbb{R}^d} q \boldsymbol{J} \right) > 0 \quad \forall q \neq 0 \quad \text{and} = 0 \text{ iff } q = 0. \quad \square
$$

Remark 1. From the Hodge decomposition (3.3a) and the divergence free condition $\nabla \cdot U^{n+1} = 0$, we have:

$$
\boldsymbol{U}^* = \boldsymbol{U}^{n+1} + \Delta t \frac{\nabla q^{n+1}}{\rho} \text{ in } \Omega^{\text{f},n+1},
$$

or

$$
\frac{\Delta t}{\rho} \Delta q^{n+1} = \nabla \cdot \boldsymbol{U}^* \text{ in } \Omega^{\text{f},n+1},
$$

which, when substituted into Eq. (3.1) for the intermediate velocity U^*, gives the following equation for the pressure update:

$$
p^{n+1} = p_d^n + \frac{3}{2} q^{n+1} - \mu \nabla \cdot \boldsymbol{U}^*. \tag{3.5}
$$

Note that this pressure update corresponds to the projection method PM II in [5] and the rotational pressure-correction method in [16]. There exists a simpler pressure update, e.g. $p^{n+1} = p_d^n + \frac{3}{2} q^{n+1}$, which corresponds to PM I in [5]. Both updates produce second-order accurate velocity field, but the pressure is second-order accurate only in the case of PM II; PM I produces only a first-order accurate pressure update [5]. In contrast with fluid simulations for which the difference in pressure accuracy between PM I and PM II is insignificant for the overall accuracy of the method, the accuracy of the fluid/rigid-body simulation depends closely on the accuracy of the pressure update. Indeed, the pressure plays an important role in the computation of the force and torque induced by the fluid stress tensor $\underline{\underline{\sigma}} = -pI + 2\mu \underline{\underline{D}}$. Therefore, we are using the pressure update given by Eq. (3.5).

4. Spatial discretization

The projection in the semi-discrete form was proven to be symmetric and positive definite using the integration by parts formula. To keep the symmetric positive definiteness in the fully discrete case, we take the standard staggered grid (MAC) arrangement [17], where the integration by parts can be enforced exactly at the discrete level. For each cell, the pressure p is sampled at its center and the velocity components are sampled on its faces, i.e. in two spatial dimensions the pressure is sampled as p_{ij} and the velocity field $\boldsymbol{U} = (u, v)$ is sampled as $u_{i\pm\frac{1}{2},j}$ and $v_{i,j\pm\frac{1}{2}}$. The rigid body geometry is represented by an implicit function ϕ, i.e. $\Gamma = \{\boldsymbol{x} : \phi(\boldsymbol{x}) = 0\}$, that is sampled at the cells' centers.

In what follows, we give the spatial discretization used in our work. Most of the spatial derivatives are discretized in a dimension by dimension fashion, in which case we describe the discretizations in two spatial dimensions only. In the case where the discretizations in three spatial dimensions are not a straightforward extension of the two dimensional case, we explicitly describe them in two as well as in three spatial dimensions.

4.1. Discretization of the gradient and divergence operators

We denote by D_x the central finite differences in the x-direction:

$$(D_x p)_{i+\frac{1}{2},j} = \frac{p_{i+1,j} - p_{i,j}}{\Delta x} \quad \text{and} \quad (D_x u)_{ij} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x}.$$

Similarly, D_y denotes the central finite difference in the y-direction. The gradient and divergence operators are also approximated by central finite differences, and are denoted by ∇^h and $\nabla^h\cdot$, respectively:

$$\nabla^h[p_{ij}] = \left(\left[(D_x p)_{i+\frac{1}{2},j}\right], \left[(D_y p)_{i+\frac{1}{2},j}\right]\right),$$
$$\nabla^h \cdot \left(\left[u_{i+\frac{1}{2},j}\right], \left[v_{i+\frac{1}{2},j}\right]\right) = \left[(D_x u + D_y v)_{ij}\right].$$

Note that the two discrete operators satisfy the integration by parts relation. Indeed consider a MAC grid with resolution $[1,M] \times [1,N]$, then we have:

$$\sum_{i=1}^{M-1}\sum_{j=1}^{N}(D_x p \cdot u)_{i+\frac{1}{2},j} + \sum_{i=1}^{M}\sum_{j=1}^{N-1}(D_y p \cdot v)_{i,j+\frac{1}{2}} = -\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}(\nabla^h \cdot \boldsymbol{U})_{ij},$$

or simply $\sum(\nabla^h p)\cdot\boldsymbol{U} = -\sum p(\nabla^h \cdot \boldsymbol{U})$ if the non-penetration boundary condition, $\boldsymbol{U}\cdot\boldsymbol{n} = 0$, or the zero pressure boundary condition, $p = 0$, are satisfied on the computational domain's boundary.

4.2. Discretization of the convection term

As already stated in Section 3.1, the convection term is discretized by a standard second-order accurate semi-Lagrangian method [36,24]. Let $\boldsymbol{x}^{n+1}$ be a grid node at time t^{n+1}, its departure point $\boldsymbol{x}_d^n$ is found by tracing back the characteristic curve using a second-order Runge–Kutta method:

$$\boldsymbol{x}^{n+\frac{1}{2}} = \boldsymbol{x}^{n+1} - \frac{\Delta t}{2}\cdot\boldsymbol{U}^n(\boldsymbol{x}^{n+1}),$$
$$\boldsymbol{x}_d^n = \boldsymbol{x}^{n+1} - \Delta t^n \cdot \boldsymbol{U}^{n+\frac{1}{2}}\left(\boldsymbol{x}^{n+\frac{1}{2}}\right),$$

The intermediate velocity $\boldsymbol{U}^{n+1/2}$ is extrapolated in time using $\boldsymbol{U}^{n+\frac{1}{2}}\left(\boldsymbol{x}^{n+\frac{1}{2}}\right) = \frac{3}{2}\boldsymbol{U}^n\left(\boldsymbol{x}^{n+\frac{1}{2}}\right) - \frac{1}{2}\boldsymbol{U}^{n-1}\left(\boldsymbol{x}^{n+\frac{1}{2}}\right)$. Finally, since the Lagrangian coordinates do not necessarily fall on grid nodes, the corresponding values for the velocity field are evaluated using the quadratic interpolation introduced in [24].

4.3. Discretization of the Heaviside function

Let H be the Heaviside function in Ω^f. In the projection step described in Section 3.4, the Heaviside function serves two purposes: first, it is used to enforce the incompressible condition $\nabla\cdot\boldsymbol{U} = 0$ in Ω^f as well as the non-penetration boundary condition $\boldsymbol{U}\cdot\boldsymbol{n} = \boldsymbol{U}^{\text{solid}}\cdot\boldsymbol{n}$ on Γ. These two conditions are combined through the divergence equation $\nabla\cdot(H(\boldsymbol{U} - \boldsymbol{U}^{\text{solid}})) = 0$. Second, the Heaviside function is used to evaluate integrals over the interface as an integral over the entire computational domain Ω : $\int_\Gamma \boldsymbol{f}\boldsymbol{n}\,d\Gamma = \int_\Omega \boldsymbol{f}\nabla H\,d\Omega$. The proper discretization of the Heaviside function is one of the most important point of this work.

There exists several efficient second-order accurate discretizations of the Heaviside function, such as those introduced in Towers [33], in Smereka [32] or in Min and Gibou [22]. These methods have been designed and validated for accurately evaluating integrals over irregular domains Ω^f as $\int_{\Omega^f}\boldsymbol{f}\,d\Omega = \int_\Omega \boldsymbol{f}H\,d\Omega$, which is one property of the Heaviside function that we wish to use. However, the use of the Heaviside function of Min and Gibou to enforce the divergence equation $\nabla\cdot(H(\boldsymbol{U} - \boldsymbol{U}^{\text{solid}})) = 0$ produces results that do not converge in L^∞-norm, as we pointed out in Ng et al. [24]. We have tested the Heaviside function introduced in Towers [33] and observed the same undesirable behavior. We have not tested the Heaviside function of [32] because it would lead to a nonsymmetric linear system for the projection step, which we want to avoid. However, we believe that such discretization would also produce results in the projection step that would not converge in the L^∞-norm. In contrast, a finite volume treatment of the standard projection step for single-phase fluid in irregular domains, has been shown to produce second-order accurate solutions in the L^∞-norm [24,27]. This implies that a correct choice for the treatment of the divergence equation $\nabla \cdot (H(\boldsymbol{U} - \boldsymbol{U}^{\text{solid}})) = 0$ is to define the Heaviside function on each cell as the length fraction (in two spatial dimensions; the area fraction in three spatial dimensions) of the face in Ω^f. This is therefore our choice in the present work and we will show in Section 4.4 the typical order of accuracy that one obtains when also using this discretization of the Heaviside function in the integration procedure.

In two spatial dimensions, consider a cell face $x_{i+\frac{1}{2},j} \times \left[y_{j-\frac{1}{2}},\ y_{j+\frac{1}{2}}\right]$ with the level function taking values $\phi_{i+\frac{1}{2},j\pm\frac{1}{2}}$ at its two end points. Assuming a linear interpolation of ϕ, the Heaviside function (length fraction) is simply computed as:

$$H_{i+\frac{1}{2},j} = \frac{\phi^+_{i+\frac{1}{2},j+\frac{1}{2}} - \phi^+_{i+\frac{1}{2},j-\frac{1}{2}}}{\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i+\frac{1}{2},j-\frac{1}{2}}},$$

where $\phi^+ = \max(\phi,0)$. In the case where both ϕ's are positive, we set $H = 1$; if both ϕ's are negative, we set $H = 0$. Thus, if the numerator equals zero, which means that the two ϕ's have the same sign, the fraction is correctly treated as 1 or 0.

In three spatial dimensions, consider a cell face $x_{i+\frac{1}{2},j,k} \times \left[y_{j-\frac{1}{2}},\ y_{j+\frac{1}{2}}\right] \times \left[z_{k-\frac{1}{2}},\ z_{k+\frac{1}{2}}\right]$ with the level function taking values $\phi_{i+\frac{1}{2},j\pm\frac{1}{2},k\pm\frac{1}{2}}$ at its four corners. We first decompose the rectangular face into two triangles and apply a linear interpolation procedure on each triangle to find their areas: Consider a triangle $\Delta P_0P_1P_2$ with level function values ϕ_0, ϕ_1, ϕ_2 at its three vertices. The area fraction covered by Ω^f is calculated as:

$$H(\phi^0,\phi^1,\phi^2) = \begin{cases} \frac{\phi_0^+ - \phi_1^+}{\phi_0 - \phi_1} \cdot \frac{\phi_0^+ - \phi_2^+}{\phi_0 - \phi_2} & \text{if } \phi^0 > 0,\ \phi^1 < 0, \quad \text{and} \quad \phi^2 < 0 \\ 1 - \frac{\phi_0^+ - \phi_1^+}{\phi_0 - \phi_1} \cdot \frac{\phi_0^+ - \phi_2^+}{\phi_0 - \phi_2} & \text{if } \phi^0 < 0,\ \phi^1 > 0, \quad \text{and} \quad \phi^2 > 0. \end{cases}$$

By symmetry, the other cases are treated similarly. The area fraction of the face occupied by Ω^f is then defined as:

$$H_{i+\frac{1}{2},j,k} = \frac{1}{2}\left(\begin{array}{l} H\left(\phi_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}, \phi_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, \phi_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right) \\ +H\left(\phi_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}, \phi_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}, \phi_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right) \end{array} \right).$$

4.4. Integral computation using the Heaviside function

As pointed out in Section 4.3, our choice for the Heaviside discretization enforces the divergence condition $\nabla \cdot (H(U - U^{solid})) = 0$ with second-order accuracy (see [24]). In this section, we provide typical accuracy results for the computation of integrals over an irregular interface as:

$$\int_\Gamma f\,d\Gamma = \int_\Omega f\|\nabla H\|\,d\Omega \simeq \sum_{ij} f_{ij}\sqrt{(D_xH)^2_{ij} + (D_yH)^2_{ij}}\Delta x\Delta y.$$

We consider the evaluation of the integral $\int_{x^2+y^2=1} 3x^2 - y^2 = 2\pi$ on a computational domain $[-2,2]^2$ in two spatial dimensions and the computation of the integral $\int_{x^2+y^2+z^2=1}(4 - 3x^2 + 2y^2 - z^2) = \frac{40}{3}\pi$ on $[-2,2]^3$. These examples are taken from [32]. Table 1 indicates that the Heaviside defined as the length fraction in two spatial dimensions approximates the integral with a convergence rate slightly superior to 1.5. Table 2 indicates that the area fraction in three spatial dimensions approximates the integral with a convergence rate of 2. The Heaviside approximations given by Towers [33] or Min and Gibou [22]

Table 1
Accuracy of the integral computation of Section 4.4 using a discretized Heaviside function in two spatial dimensions (length L).

grid	$\|L - L_{\text{exact}}\|_\infty$	order
40^2	3.74×10^{-2}	
80^2	1.23×10^{-2}	1.60
160^2	4.37×10^{-3}	1.49
320^2	1.50×10^{-3}	1.54
640^2	5.36×10^{-4}	1.48

Table 2
Accuracy of the integral computation of Section 4.4 using the discretized Heaviside function in three spatial dimensions (area A).

grid	$\|A - A_{\text{exact}}\|_\infty$	order
20^3	1.01×10^{0}	
40^3	2.27×10^{-1}	2.15
80^3	5.43×10^{-2}	2.06
160^3	1.32×10^{-2}	2.04

produce second-order accurate results in both two and three spatial dimensions. Of course, it would be conceivable to use those approximations for the discretization of the integral part only, but this would lead to a non-symmetric linear system in the projection step, which we want to avoid. We will show in Section 7 that our choice of approximations of the Heaviside function produces second-order accurate solutions in the L^∞-norm for the fluid/solid interactions.

4.5. Discretization of the diffusion term

The combination of the semi-Lagrangian and BDF schemes for approximating the intermediate velocity field in the Navier–Stokes equations reads:

$$\rho\frac{\frac{3}{2}\boldsymbol{U}^* - 2\boldsymbol{U}_d^n + \frac{1}{2}\boldsymbol{U}_d^{n-1}}{\Delta t} + \nabla p_d^n = \mu\Delta\boldsymbol{U}^* + \boldsymbol{g} \text{ in } \Omega^{f,n+1},$$
$$\boldsymbol{U}^* = \boldsymbol{U}^{solid,n+1} \text{ on } \Gamma^{n+1}.$$

We considered two methodologies for treating implicitly the diffusion tensor with Dirichlet boundary condition: The symmetric discretization of Gibou et al. [14] and the non-symmetric discretization of Shortley and Weller [31,13]. Both methods produce second-order accurate solutions. However, only the non-symmetric discretization produces second-order accurate gradients; the symmetric approach produces first-order accurate gradients only [23,19]. In the case of fluid/solid coupling, the traction force $(-p\underline{\underline{I}} + \mu(\nabla\boldsymbol{U} + (\nabla\boldsymbol{U})^T))\cdot\boldsymbol{n}$ requires an accurate evaluation of $\nabla\boldsymbol{U}$, the gradient of the solution. For this reason, we discretize the diffusion term $\mu\Delta\boldsymbol{U}^*$ by the non-symmetric discretization, and solve the linear system by the BiCGSTAB iteration with incomplete LU preconditioning [30].

4.6. Discretization of the projection step

In Section 3.4, we derived the following equation for the scalar q^{n+1} used in the projection step:

$$-\nabla\cdot\left(\frac{H}{\rho}\nabla q\right) + \nabla H\cdot\frac{1}{m}\left(\int q\nabla H\right) + \boldsymbol{J}\cdot\underline{\underline{I}}^{-1}\left(\int q\boldsymbol{J}\right) = -\frac{1}{\Delta t}\left(\nabla\cdot(H\boldsymbol{U}^*) + \nabla H\cdot\boldsymbol{U}^{\text{solid},*}\right),$$

where we omitted the superscript ${}^{n+1}$ for convenience. All the spatial derivatives in equation are approximated by central finite differences as discussed in Section 4.1, the Heaviside function is approximated by the finite volume discretization described in Section 4.3 and the integrals are approximated as detailed in Section 4.4. For each grid index i and j, we then have:

$$-\left(\nabla^h\cdot\left(\frac{H}{\rho}\nabla^h q\right)\right)_{i,j} + (\nabla^h H)_{i,j}\cdot\frac{1}{m}\left(\sum_{k,l}(q\nabla^h H)_{k,l}dV\right) + \boldsymbol{J}_{i,j}\cdot\underline{\underline{I}}^{-1}\left(\sum_{k,l}(q\boldsymbol{J})_{k,l}dV\right) = -\frac{1}{\Delta t}(\nabla^h\cdot(H\boldsymbol{U}^* - H\boldsymbol{U}^{solid,*}))_{i,j}, \qquad (4.1)$$

where dV denotes the volume element $\Delta x\cdot\Delta y$.

Theorem 2. *The linear system associated with the fully discrete equation (4.1) is symmetric and positive definite.*

Proof. Let $\mathfrak{L}^{\mathfrak{h}}(\cdot)$ be the discrete operator on the left-hand-side of (4.1) acting on $q_{i,j}^{n+1}$. For arbitrary $\left[q_{ij}^{(1)}\right]$ and $\left[q_{ij}^{(2)}\right]$, we have:

$$\begin{aligned}\left[q_{ij}^{(1)}\right]\cdot[(\mathfrak{L}^{\mathfrak{h}}(q^{(2)}))_{ij}] &= -\sum_{i,j}q_{i,j}^{(1)}\left(\nabla^h\cdot\left(\frac{H}{\rho}\nabla^h q^{(2)}\right)\right)_{i,j} + \sum_{i,j}q_{ij}^{(1)}(\nabla^h H)_{i,j}\cdot\frac{1}{m}\left(\sum_{k,l}(q^{(2)}\nabla^h H)_{k,l}dV\right) + \sum_{i,j}(q^{(1)}\boldsymbol{J})_{i,j}\\ &\quad\cdot\underline{\underline{I}}^{-1}\left(\sum_{k,l}(q^{(2)}\boldsymbol{J})_{k,l}dV\right)\\ &= -\sum_{i,j}\frac{H_{i,j}}{\rho}(\nabla^h q^{(1)})_{i,j}(\nabla^h q^{(2)})_{i,j} + \frac{dV}{m}\left(\sum_{i,j}(q^{(1)}\nabla^h H)_{i,j}\right)\cdot\left(\sum_{k,l}(q^{(2)}\nabla^h H)_{k,l}\right)\\ &\quad+\left(dV\cdot\underline{\underline{I}}^{-1}\left(\sum_{i,j}(q^{(1)}\boldsymbol{J})_{i,j}\right)\right)\cdot\left(\sum_{k,l}(q^{(2)}\boldsymbol{J})_{k,l}\right).\end{aligned}$$

Note that we can apply the integration by parts formula of Section 4.1, since $\nabla^h q\cdot\mathbf{n} = 0$ is assumed at the boundary of the computational domain. This is clearly a symmetric form for $q^{(1)}$ and $q^{(2)}$, since the inertia matrix and its inverse are symmetric. The positive definiteness of the operator $\mathfrak{L}^h$ is a straightforward consequence of the fact that the inertia matrix is positive definite. For any L^2-integrable function q, we have:

$$\begin{aligned}[q_{ij}]\cdot[(\mathfrak{L}^{\mathfrak{h}}(q))_{ij}] &= -\sum_{i,j}\frac{H_{i,j}}{\rho}\|\nabla^h q\|_{i,j}^2 + \frac{dV}{m}\left(\sum_{i,j}(q\nabla^h H)_{i,j}\right)^2 + \left(dV\cdot\underline{\underline{I}}^{-1}\left(\sum_{i,j}(q\boldsymbol{J})_{i,j}\right)\right)\cdot\left(\sum_{k,l}(q\boldsymbol{J})_{k,l}\right)\\ &> 0 \quad \forall\, q \neq 0 \quad \text{and} = 0 \text{ iff } q = 0. \quad \square\end{aligned}$$

Remark 2. It is clear that if the Heaviside functions of [22,33,32] were used as discussed in Section 4.3, the linear system would not be symmetric.

Finally, once q is solved the states variables ($\boldsymbol{U}^*$,$\boldsymbol{P}^*$,$\boldsymbol{L}^*$) are projected as:

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^* - \Delta t\frac{\nabla^h q}{\rho},$$

$$\boldsymbol{P}^{n+1} = \boldsymbol{P}^* - \Delta t\sum_{i,j}(q\nabla^h H)_{i,j}dV,$$

$$\boldsymbol{L}^{n+1} = \boldsymbol{L}^* - \Delta t\sum_{i,j}(q\boldsymbol{J})_{i,j}dV.$$

The following theorem demonstrates that the projection is stable in the sense that it does not increase the kinetic energy:

Theorem 3. *The projection step does not increase the kinetic energy of the fluid and the solid.*

Proof. The kinetic energy of the fluid and the solid is defined as $K = \int_\Omega \frac{\rho}{2}\boldsymbol{U}^2 + \frac{1}{2m}\boldsymbol{P}\cdot\boldsymbol{P} + \frac{1}{2}\underline{\underline{I}}^{-1}\boldsymbol{L}\cdot\boldsymbol{L}$, and discretized as:

$$K(\boldsymbol{U},\boldsymbol{P},\boldsymbol{L}) = \frac{1}{2}[\rho H\boldsymbol{U}]\cdot[\boldsymbol{U}] + \frac{1}{2m}\boldsymbol{P}\cdot\boldsymbol{P} + \frac{1}{2}\underline{\underline{I}}^{-1}\boldsymbol{L}\cdot\boldsymbol{L},$$

where $[\boldsymbol{U}^{(1)}]\cdot[\boldsymbol{U}^{(2)}] = \sum_{ij}u^{(1)}_{i+\frac{1}{2},j}u^{(2)}_{i+\frac{1}{2},j}dV + \sum_{ij}v^{(1)}_{i,j+\frac{1}{2}}v^{(2)}_{i,j+\frac{1}{2}}dV$ denotes the inner product for the vector fields $U^{(1)} = (u^{(1)}, v^{(1)})$ and $U^{(2)} = (u^{(2)}, v^{(2)})$. Similarly, $[q^{(1)}]\cdot[q^{(2)}] = \sum_{ij}q^{(1)}_{ij}q^{(2)}_{ij}dV$ denotes the inner product between scalar quantities $q^{(1)}$ and $q^{(2)}$. Combining $K(U^*,P^*,L^*)$ with Eqs. (3.3a), (3.3b), and (3.3c) we have:

$$\begin{aligned} K(\boldsymbol{U}^*,\boldsymbol{P}^*,\boldsymbol{L}^*) = {} & K(\boldsymbol{U}^{n+1},\boldsymbol{P}^{n+1},\boldsymbol{L}^{n+1}) + \Delta t\cdot[\rho H\boldsymbol{U}^{n+1}]\cdot\left[\frac{\nabla^h q}{\rho}\right] + \Delta t\cdot\left(\boldsymbol{P}^{n+1}\cdot\frac{1}{m}(\sum_{ij}(q\nabla^h H)_{ij}dV) + \boldsymbol{L}^{n+1}\cdot\underline{\underline{I}}^{-1}(\sum_{ij}(q\boldsymbol{J})_{ij}dV)\right) \\ & + \frac{\Delta t^2}{2}\left(\left[\rho H\frac{\nabla^h q}{\rho}\right]\cdot\left[\frac{\nabla^h q}{\rho}\right]\right) + \frac{\Delta t^2}{2}\left(\frac{1}{m}(\sum_{ij}(q\nabla^h H)_{ij}dV)^2 + \underline{\underline{I}}^{-1}(\sum_{ij}(q\boldsymbol{J})_{ij}dV)^2\right) \\ & \geqslant K(\boldsymbol{U}^{n+1},\boldsymbol{P}^{n+1},\boldsymbol{L}^{n+1}) + \Delta t\cdot[H\boldsymbol{U}^{n+1}]\cdot[\nabla^h q] + \Delta t\cdot\left(\boldsymbol{P}^{n+1}\cdot\frac{1}{m}\left(\sum_{ij}(q\nabla^h H)_{ij}dV\right) + \boldsymbol{L}^{n+1}\cdot\underline{\underline{I}}^{-1}(\sum_{ij}(q\boldsymbol{J})_{ij}dV)\right). \end{aligned}$$

Using the discrete integration by parts formula on MAC grids (see Section 4.1), we have:

$$\begin{aligned} [H\boldsymbol{U}^{n+1}]\cdot[\nabla^h q] & = -[\nabla^h\cdot(H\boldsymbol{U}^{n+1})]\cdot[q] = -[\nabla^h H\cdot\boldsymbol{U}^{\text{solid},n+1}]\cdot[q] = -\sum_{ij}\left(q\nabla^h H\cdot\left(\frac{1}{m}\boldsymbol{P}^{n+1} + \underline{\underline{I}}^{-1}\boldsymbol{L}^{n+1}\times(\boldsymbol{x}-\boldsymbol{C}^{n+1})\right)\right)_{ij}dV \\ & = -P^{n+1}\cdot\frac{1}{m}\left(\sum_{ij}(q\nabla^h H)_{ij}dV\right) - L^{n+1}\cdot\underline{\underline{I}}^{-1}\left(\sum_{ij}(q\boldsymbol{J})_{ij}dV\right). \end{aligned}$$

Thus we obtain that $K(U^*,P^*,L^*) \geqslant K(U^{n+1},P^{n+1},L^{n+1})$. □

5. Preconditioning the SPD linear system of the projection step

If we denote by $\underline{\underline{\boldsymbol{A}}}$ the matrix associated with the standard discretization of the Poisson operator $-\nabla^h\cdot\left(\frac{H^{n+1}}{\rho}\nabla^h\right)$, then the matrix of the linear system for the projection step of the fluid/solid coupling problem at hand is defined as:

$$\underline{\underline{\boldsymbol{A}}}^{\text{FSC}} := \underline{\underline{\boldsymbol{A}}} + (\nabla^h H)\cdot\frac{dV}{m}(\nabla^h H)^T + \boldsymbol{J}\cdot dV\cdot\underline{\underline{I}}^{-1}\boldsymbol{J}^T.$$

Let $\underline{\underline{\boldsymbol{M}}}$ be the modified incomplete LU preconditioner for $\underline{\underline{\boldsymbol{A}}}$. Since $\underline{\underline{\boldsymbol{M}}} \simeq \underline{\underline{\boldsymbol{A}}}$, we take the preconditioner for the matrix $\underline{\underline{\boldsymbol{A}}}^{\text{FSC}}$ as:

$$\underline{\underline{\boldsymbol{M}}}^{\text{FSC}} := \underline{\underline{\boldsymbol{M}}} + (\nabla^h H)\cdot\frac{dV}{m}(\nabla^h H)^T + \boldsymbol{J}\cdot dV\cdot\underline{\underline{I}}^{-1}\boldsymbol{J}^T.$$

Note that since $\nabla^h H$ and $\boldsymbol{J}$ have support only near the interface, they are stored as sparse matrices for the sake of preserving computational resources.

Now we derive a formula for the inversion of the system $\underline{\underline{\boldsymbol{M}}}^{\text{FSC}}\boldsymbol{q_h} = \boldsymbol{RHS}$, where $\boldsymbol{q_h}$ is the vector whose components are the $q_{i,j}$'s and $\boldsymbol{RHS}$ the vector associated with the sampling of the right-hand-side of Eq. (4.1). It is convenient to write equation in the following block linear system, which is then solved by a standard block LU factorization:

$$\begin{pmatrix} \underline{\underline{\boldsymbol{M}}} & \nabla^h H & \boldsymbol{J} \\ (\nabla^h H)^T & -\frac{m}{dV}\underline{\underline{\boldsymbol{I}}} & \underline{\underline{\boldsymbol{0}}} \\ \boldsymbol{J}^T & \underline{\underline{\boldsymbol{0}}} & -\frac{1}{dV}\cdot\underline{\underline{I}}^{-1}\end{pmatrix}\begin{pmatrix}\boldsymbol{q_h}\\ \boldsymbol{v}\\ \omega\end{pmatrix} = \begin{pmatrix}\boldsymbol{RHS}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{pmatrix},$$

$$\iff \begin{pmatrix} \underline{\underline{\boldsymbol{I}}} & \underline{\underline{\boldsymbol{0}}} & \underline{\underline{\boldsymbol{0}}} \\ (\nabla^h H)^T\underline{\underline{\boldsymbol{M}}}^{-1} & \underline{\underline{\boldsymbol{I}}} & \underline{\underline{\boldsymbol{0}}} \\ \boldsymbol{J}^T\underline{\underline{\boldsymbol{M}}}^{-1} & \boldsymbol{U_1}\underline{\underline{\boldsymbol{D_1}}}^{-1} & \underline{\underline{\boldsymbol{I}}}\end{pmatrix}\begin{pmatrix} \underline{\underline{\boldsymbol{M}}} & \nabla^h H & \boldsymbol{J} \\ \underline{\underline{\boldsymbol{0}}} & \underline{\underline{\boldsymbol{D_1}}} & \underline{\underline{\boldsymbol{U_1}}} \\ \underline{\underline{\boldsymbol{0}}} & \underline{\underline{\boldsymbol{0}}} & \underline{\underline{\boldsymbol{D}}}_2\end{pmatrix}\begin{pmatrix}\boldsymbol{q_h}\\ \boldsymbol{v}\\ \omega\end{pmatrix} = \begin{pmatrix}\boldsymbol{RHS}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{pmatrix},$$

The block matrices $\underline{\underline{\boldsymbol{D}}}_1$, $\underline{\underline{\boldsymbol{D}}}_2$, and $\underline{\underline{\boldsymbol{U}}}_1$ are of size 3×3 in three spatial dimensions, and of size 2×2, 1×1 and 2×1, respectively in two spatial dimensions. The matrices are defined as $\underline{\underline{\boldsymbol{D}}}_1 = -\frac{m}{dV}\underline{\underline{\boldsymbol{I}}} - (\nabla^h H)^T\underline{\underline{\boldsymbol{M}}}^{-1}(\nabla^h H)$, $\underline{\underline{\boldsymbol{U}}}_1 = -\boldsymbol{J}^T\underline{\underline{\boldsymbol{M}}}^{-1}(\nabla^h H)$, and $\underline{\underline{\boldsymbol{D}}}_2 = -\frac{1}{dV}\cdot\underline{\underline{I}}^{-1} - \boldsymbol{J}^T\underline{\underline{\boldsymbol{M}}}^{-1}\boldsymbol{J} - \underline{\underline{\boldsymbol{U}}}_1\underline{\underline{\boldsymbol{D_1}}}^{-1}\underline{\underline{\boldsymbol{U}}}_1$. These matrices are evaluated and stored, whereas $\underline{\underline{\boldsymbol{M}}}^{-1}$ is not stored but only applied when needed as for fluid-only solvers [15].

The MILU preconditioner $\underline{\underline{\boldsymbol{M}}}$ is known to be a very efficient preconditioner for the matrix $\underline{\underline{\boldsymbol{A}}}$ associated with standard fluid-only simulations. In order to quantify the performance of our preconditioner $\underline{\underline{\boldsymbol{M}}}^{\mathrm{FSC}}$ for the matrix $\underline{\underline{\boldsymbol{A}}}^{\mathrm{FSC}}$, we consider the following example in two spatial dimensions: Consider a fluid with uniform intermediate velocity field $\boldsymbol{U}^* = (0,-1)$ on a computational domain $[-0.02, 0.02]^2$. A cylindrical solid object with center (0,0) and radius $r = 0.005$ is immersed in the surrounding fluid. The densities of the fluid and the solid are set to be $\rho_f = 1000$ and $\rho_s = 2000$, respectively. The solid has linear momentum $\boldsymbol{P}^* = \rho_s\pi r^2(0,-1)$ and angular momentum $\boldsymbol{L}^* = 0$. Non-slip boundary conditions are imposed on the boundary of the computational domain. The fluid and solid velocity fields are projected by solving the linear system $\underline{\underline{\boldsymbol{A}}}^{\mathrm{FSC}}\boldsymbol{q_h} = \boldsymbol{RHS}$.

Fig. 5.1 illustrates the reduction in the number of iterations between the CG and the PCG algorithms in the case of the standard fluid-only and the present fluid/solid projection methods on a grid with resolution 500^2. These results indicate that the computational complexity of the projection step of our fluid/solid coupling is on a par with that of a standard fluid-only solver. In particular, in the case of the conjugate gradient, the norm of the residual $|r_n|$ in log scale decreases linearly with the number of iterations n with slope $-\frac{2}{\sqrt{\kappa}}$, where κ is the condition number of the matrix [34]. Fig. 5.1(a) compares the CG and PCG iterations for the standard projection method; the slope for the CG iterations is about -0.015 and that of PCG is about -0.066, which means that the $\underline{\underline{\boldsymbol{M}}}$ preconditioner reduces the condition number of $\underline{\underline{\boldsymbol{A}}}$ by a factor $\left(\frac{-0.066}{-0.015}\right)^2 \simeq 19$. Likewise, Fig. 5.1(b) compares the CG and PCG iterations for the present projection method; the slope for the CG iteration is about -0.016 and that of PCG is about -0.074, which means that the $\underline{\underline{\boldsymbol{M}}}^{\mathrm{FSC}}$ preconditioner reduces the condition number of $\underline{\underline{\boldsymbol{A}}}^{\mathrm{FSC}}$ by a factor $\left(\frac{-0.074}{-0.016}\right)^2 \simeq 21$. Fig. 5.2 illustrates that the number of iterations and the computational time of the present projection step grow at a similar rate as those of the standard projection under grid refinement. This further indicates that the computational complexity of the projection step of our fluid/solid coupling is on a par with that of a standard fluid-only solver.

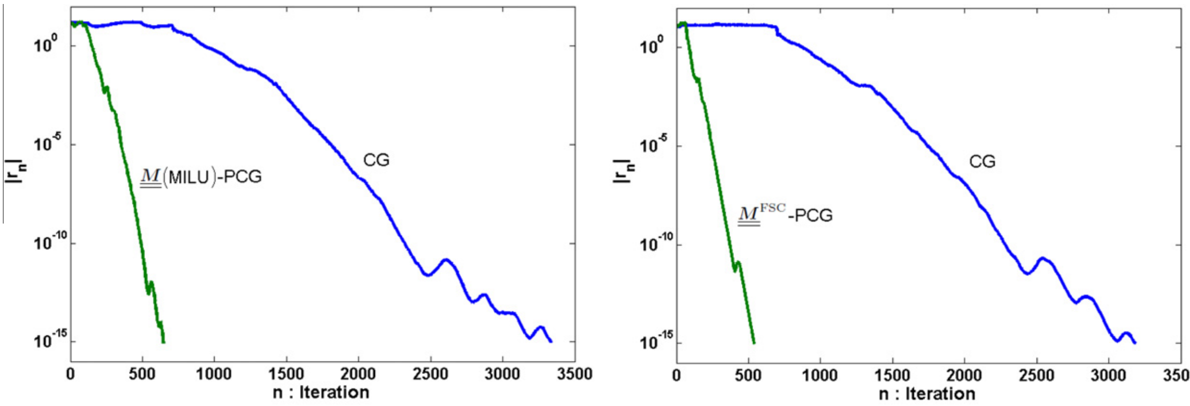
6. Summary: outline of the method

For each time step t^n, the following steps are sequentially carried out:

Step 1. Solve for $\boldsymbol{C}^{n+1}$, $\boldsymbol{P}^*$, $\boldsymbol{L}^*$, and $\underline{\underline{\boldsymbol{R}}}^{n+1}$ using the evolution equation (3.2).

Step 2. Calculate $\phi^{n+1}(\boldsymbol{x}) = \phi^0(\underline{\underline{\boldsymbol{R}}}^{n+1}(\underline{\underline{\boldsymbol{R}}}^0)^{-1}(\boldsymbol{x} - \boldsymbol{C}^0) + \boldsymbol{C}^{n+1})$ and H^{n+1}, as described in Section 4.3.

Step 3. Calculate $\boldsymbol{U}_d^n, \boldsymbol{U}_d^{n-1}$, and p_d^n via the semi-Lagrangian method described in Section 4.2.



(a) Standard fluid simulations

(b) Present fluid/solid coupling

Fig. 5.1. Convergence history of the residual $|r_n|$ for the CG and PCG algorithms in the case of the projection step. Standard fluid projection (left) and the present fluid/solid interaction algorithm (right).

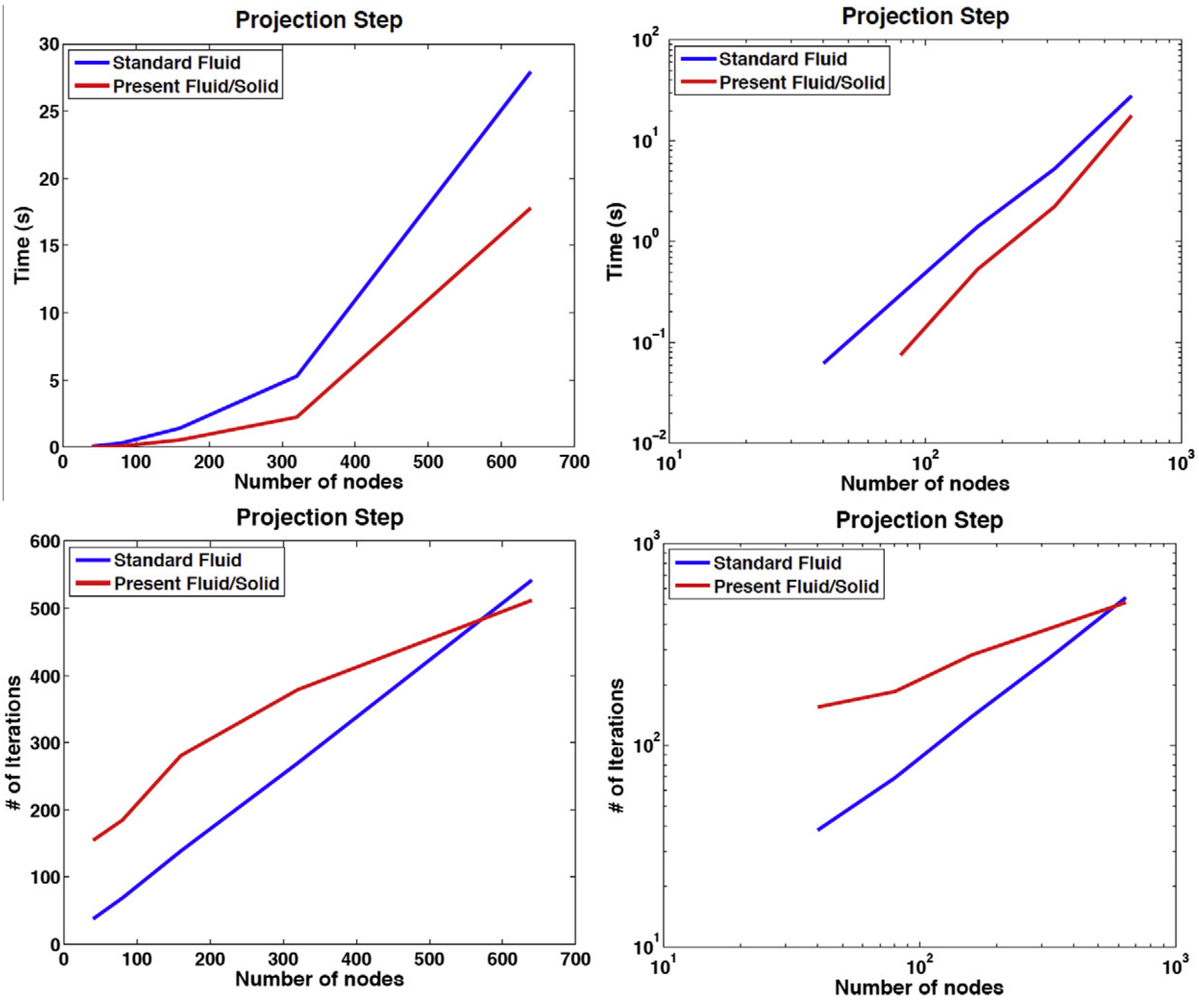


Fig. 5.2. Computational resources necessary in the projection step versus the number of nodes (the right column is in log–log scale). The PCG routine for each case was iterated until the residual $|r_n| < 10^{-12}$.

Step 4. Solve for $\boldsymbol{U}^*$ in Eq. (3.1).
Step 5. Solve for q^{n+1} in Eq. (4.1).
Step 6. Perform the projection of $\boldsymbol{U}^*$, $\boldsymbol{P}^*$ and $\boldsymbol{L}^*$:

$$\begin{aligned}
\boldsymbol{U}^{n+1} &= \boldsymbol{U}^* - \Delta t \frac{\nabla^h q}{\rho}, \\
\boldsymbol{P}^{n+1} &= \boldsymbol{P}^* - \Delta t \sum_{i,j} (q \nabla^h H)_{i,j} dV, \\
\boldsymbol{L}^{n+1} &= \boldsymbol{L}^* - \Delta t \sum_{i,j} (q\boldsymbol{J})_{i,j} dV.
\end{aligned}$$

Step 7. Extrapolate $\boldsymbol{U}^{n+1}$ and q^{n+1} from $\Omega^{f,n+1}$ to the entire computational region using Aslam's quadratic extrapolation procedure [2,21].
Step 8. Update $p^{n+1} = p_d^n + \frac{3}{2} q^{n+1} - \mu \nabla^h \cdot \boldsymbol{U}^*$.
Step 9. Go back to Step 1.

7. Numerical examples

In this section, we provide numerical evidence that our method is second-order accurate in the L^∞-norm and can reproduce known experimental results.

7.1. Sliding rectangular structure

We consider the analytical example proposed in [28], where a solid rectangular structure immersed in fluid is flowing along the center of an infinite channel under the action of gravity. A simple analytical formula is derived in [28] for the settling velocity of the solid:

$$v_s = -(M + \rho w h)\frac{gw}{2h\mu}, \tag{7.1}$$

where $h = 1$ m is the height of the channel, $w = 1/3$ m is the width of the rigid body, $M = 150$ kg its mass, $g = 9.8$ m s^{-1} is the gravity constant and $\rho = 100$ kg m^{-2} and $\mu = 100$ kg s^{-1} are the fluid's density and viscosity, respectively. The analytic velocity profile for the left fluid region $[0,w] \times [0,h]$ is $u = 0$ and $v = \frac{\rho g}{2\mu}x(x-w) + \frac{v_s}{w}x$, therefore a quadratic form in x. Fig. 7.1 depicts the convergence of our numerical solution to the analytical formula (7.1) with $\Delta t = \Delta x$. Table 3 demonstrates the second-order convergence for solid velocity at $t = 0.5$. The error at steady-state ($t = 40$) is near machine precision, which is characteristic of a second-order accurate approximation to a quadratic polynomial.

7.2. Falling cylinder

Consider the classical problem of an infinite solid cylinder with density ρ_s falling in a fluid with density ρ_f and viscosity μ, enclosed in a channel. In the case where the infinite section of the cylinder is perpendicular to the direction of motion, this problem can be treated in a two-dimensional setting. We take the dimensions of the channel to be $2L \times 8L$ and the radius of the cylinder to be r. Balancing the frictional force on the cylinder by the force due to the difference of the weight of the cylinder and its buoyancy, an analytical form for the terminal velocity can be derived using the Stokes assumption of low Reynolds number [35]:

$$v_{\text{terminal}} = \frac{(\rho_s - \rho_f)gr^2}{4\mu}\left(-\ln\left(\frac{r}{L}\right) - .9157 + 1.7244\left(\frac{r}{L}\right)^2 - 1.7302\left(\frac{r}{L}\right)^4\right) \tag{7.2}$$

We take $L = 2 \times 10^{-2}$ m, $r = 5 \times 10^{-3}$ m, $\rho_s = 2 \times 10^3$ kg m^{-2}, $\rho_f = 1 \times 10^3$ kg m^{-2} and $g = 9.8$ m s^{-1}. We apply no-slip boundary conditions at the walls of the channel and apply an outflow boundary condition at the top of the channel. The time step is $\Delta t = 2\Delta x$. Fig. 7.2 depicts our numerical results for the vertical velocity of the cylinder for different values of the fluid's viscosity $\mu = 0.05$ kg s^{-1}, $\mu = 0.1$ kg s^{-1}, $\mu = 0.2$ kg s^{-1} and $\mu = 0.5$ kg s^{-1}. These values for the viscosity correspond to Reynolds numbers of $Re \approx 140.18$, $Re \approx 35$, $Re \approx 8.74$ and $Re \approx 1.4$. Here, we take $Re = \frac{\rho V d}{\mu}$ with the characteristic velocity V to be the analytical terminal velocity and the characteristic length $d = 2r$. It is well known that the Stokes assumption is valid for

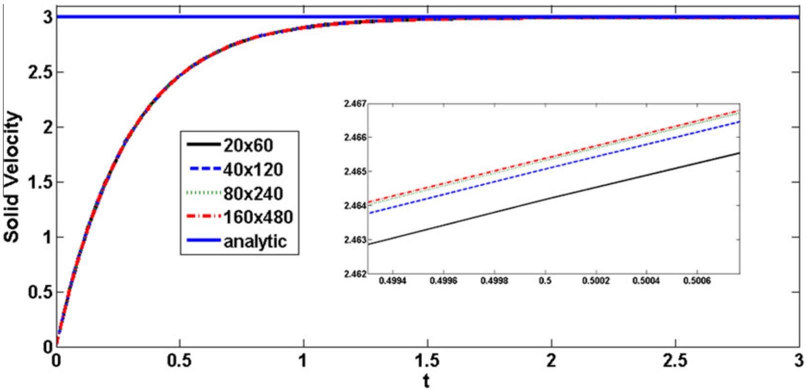
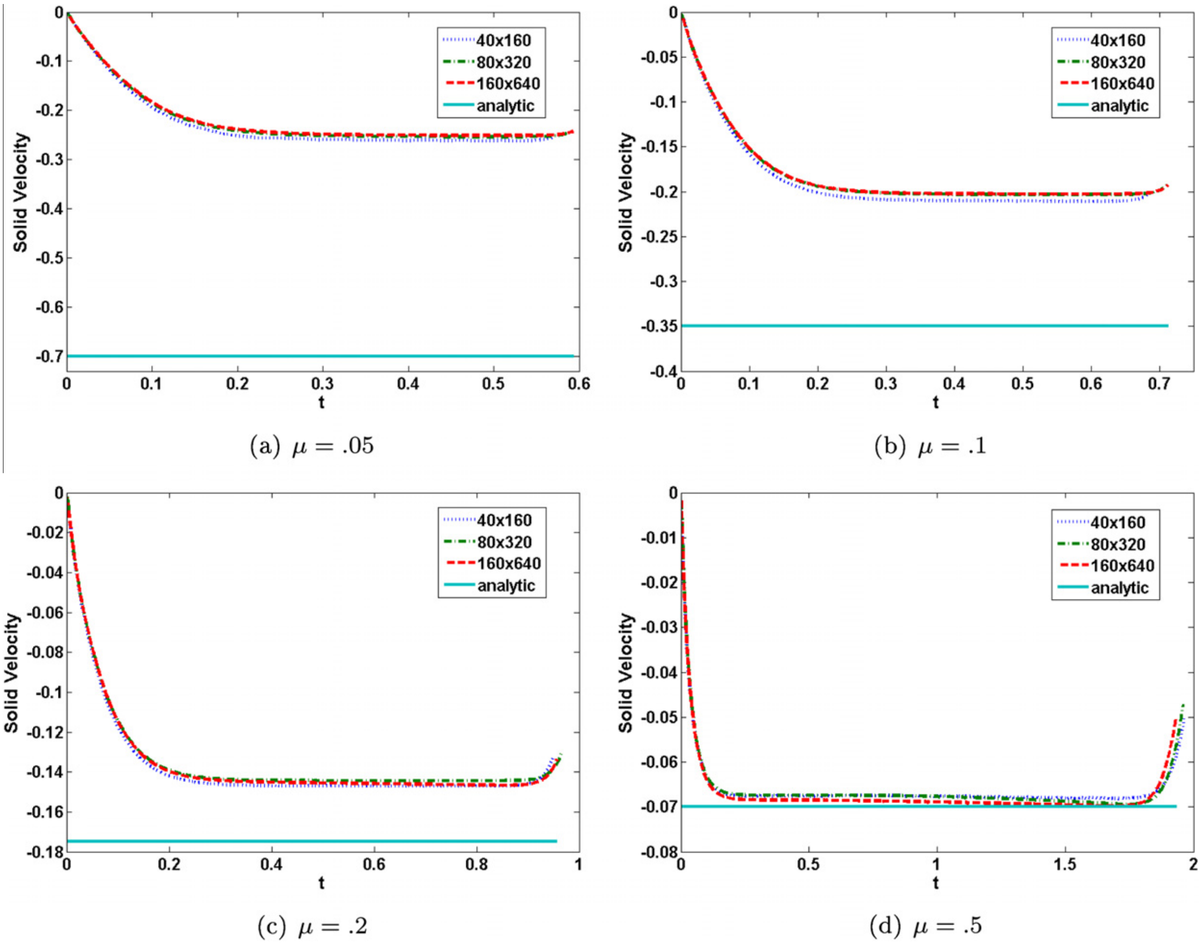


Fig. 7.1. Convergence of the solid velocity in the case of the analytical example of [28].

Table 3
Accuracy of the vertical component of the velocity field of solid in the example of Section 7.1. For calculating the error, the analytic formula for the settling velocity $v_s = -(M + \rho w h)\frac{gw}{2h\mu}$ was used at steady-state ($t = 40$) and the numerical solution on a very fine 1280^2 grid was used in the transient region ($t = 0.5$).

grid resolution	$t = 0.5$		$t = 40$	
	$\lVert v - v_{1280}\rVert_\infty$	rate	$\lVert v - v_s\rVert_\infty$	rate
20^2	1.22×10^{-3}		8.05×10^{-9}	
40^2	3.32×10^{-4}	1.88	6.49×10^{-9}	-
80^2	8.68×10^{-5}	1.93	5.94×10^{-9}	-
160^2	2.16×10^{-5}	2.00	9.34×10^{-9}	-



(a) $\mu = .05$

(b) $\mu = .1$

(c) $\mu = .2$

(d) $\mu = .5$

Fig. 7.2. Velocity of a falling cylinder in a channel over time (dashed lines) compared with the theoretical terminal velocity obtained under the Stokes assumption (solid lines).

Reynolds number $Re \lessapprox 0.1$ and in that case the terminal velocity formula given by (7.2) is accurate to within about 1%. However, the increasing effects of inertia invalidate (7.2) for increasing value of Re. This explains the deviation from the analytical prediction. Our results are consistent with the results of [28] who also pointed out the source of the deviation.

In terms of accuracy analysis, we first follow the work of Apte et al. [1], who performed a standard convergence analysis by comparing their numerical results to a reference solution obtained by their numerical method on a more refined grid. Fig. 7.3 illustrates such a convergence analysis for our method, in which the numerical solutions obtained on a 320 × 1280 grid is taken as the reference solution and where numerical solutions on grid resolutions of 40 × 160, 60 × 240, 80 × 320, 120 × 480, 160 × 640 and 180 × 720 are used for the analysis. This analysis, however, can be misleading in the case where the reference solution is not computed on a significantly more refined grid than the other solutions. In this case, the order of accuracy is overestimated since the results from successive computations get closer and closer to the reference solution. For example in our case, such an analysis indicates a convergence rate of 2.89, as illustrated in Fig. 7.3 (top-right), which is clearly excessive for an algorithm with second-order accurate approximations all around. In fact, Table 4 shows a second-order accuracy when the numerical solutions on rather coarse 40 × 160, 60 × 240 and 80 × 320 grids are compared to a reference solution on a relatively fine grid of 320 × 1280, whereas the accuracy is overestimated to third-order accuracy as the numerical solutions are computed on finer grids of 120 × 480, 160 × 640 and 180 × 720.

A more accurate procedure to measure the rate of convergence is as follows: Let p be the order of convergence of our numerical method, then the numerical solution v satisfies $v = v_{\text{exact}} + C(\Delta x)^p$ for some constant C. Writing this equation in the case where we take 320 grid nodes in the x-direction, we have $v_{320} = v_{\text{exact}} + C(\Delta x_{320})^p$. The measured error with the reference solution then satisfies $e = C((\Delta x)^p - (\Delta x_{320})^p)$. The constants C and p can then be extracted from the available data. In our case, we find $C = 2.02 \times 10^4$ and $p = 2.02$ using the nonlinear least-square fit command `nlinfit` in MATLAB, as illustrated in Fig. 7.3 (bottom). This analysis confirms that our numerical method is second-order accurate in the maximum norm.

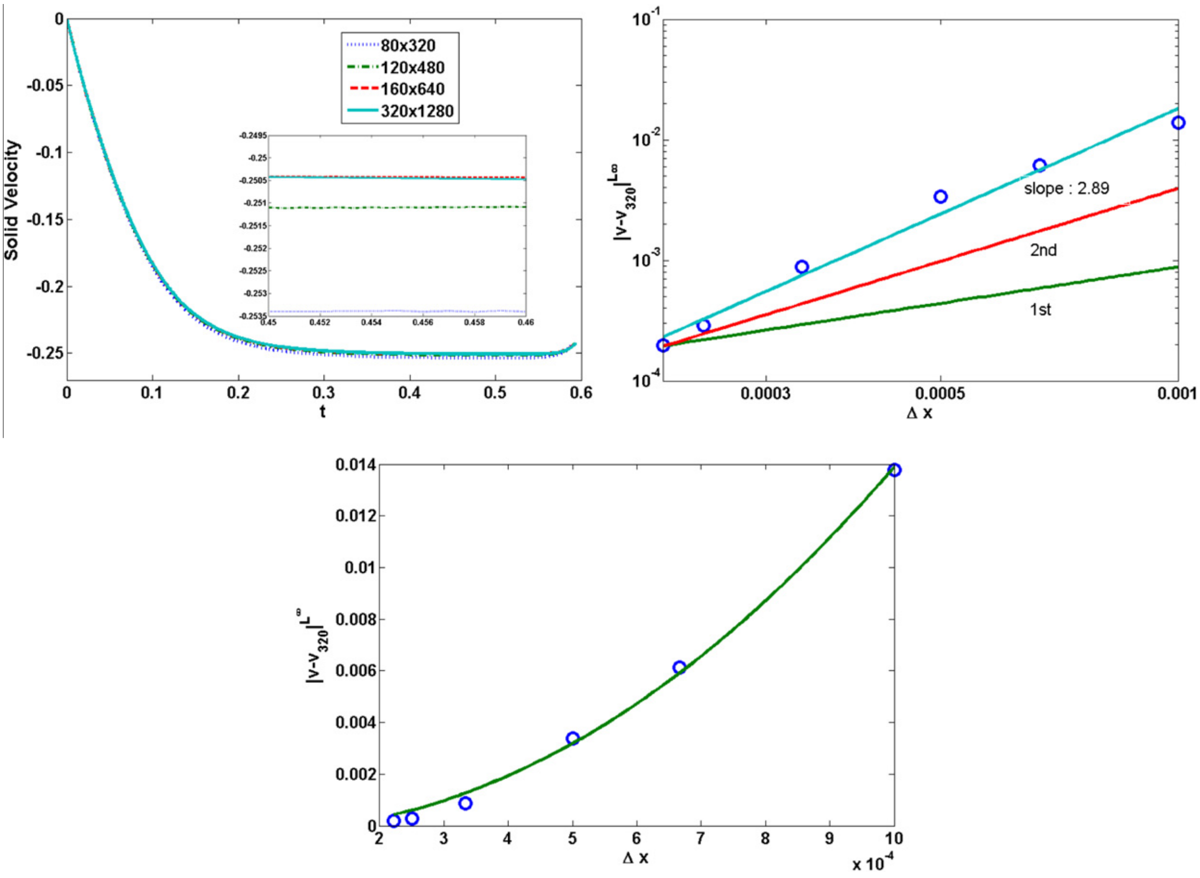


Fig. 7.3. Convergence study for the falling cylinder example of Section 7.2 with $\mu = 0.05$ kg s^{-1}. The top-left figure depicts the convergence under grid refinement. The top-right figure depicts the error versus Δx in a log–log scale when the numerical solution on a 320 × 1280 grid is used as the reference solution. In this case the order of accuracy is found to be 2.89, which is misleading. The bottom figure depicts the curve fitting of the error $e = C((\Delta x)^p - (\Delta x_{320})^p)$ with $C = 2.02 \times 10^4$ and $p = 2.05$. Here the order of accuracy is found to be 2.05. In each case, the error is measured over the time interval $t \in [0, 0.56]$ in the L^∞-norm.

Table 4
Convergence study for the falling cylinder example of Section 7.2. The convergence rate for the nth row is computed as $\log_{\left|\frac{\Delta x_{n-1}}{\Delta x_n}\right|}\left|\frac{v_{n-1}-v_{\text{ref}}}{v_n-v_{\text{ref}}}\right|$, where v_{ref} is the computed solution on a fine 320 × 1280 grid.

grid	$\|v - v_{320\times1280}\|_\infty$	rate
40 × 160	1.38×10^{-2}	
60 × 240	6.14×10^{-3}	1.99
80 × 320	3.38×10^{-3}	2.07
120 × 480	8.80×10^{-4}	3.31
160 × 640	2.88×10^{-4}	3.88
180 × 720	1.95×10^{-4}	3.31

7.3. Falling sphere

Consider a container with dimension 10 × 10 × 16 cm^3 filed with a fluid with density ρ_f and viscosity μ_f. A solid particle with diameter $d_p = 15$ mm and density $\rho_s = 1120$ kg m^{-3} is released from a height of $H = 12$ cm from the bottom of the tank and falls under the action of gravity. The fluid characteristics are varied to obtain different Reynolds number $Re = \frac{\rho_f u_\infty d_p}{\mu_f}$ based on the terminal velocity u_∞ of the particle, as detailed in Table 5. No slip boundary conditions are applied on all the walls of the container. Neumann boundary conditions for the velocity components are imposed on the top of the tank. We apply the algorithm described in this paper with grid spacings $\Delta x = \Delta y = \Delta z$ and a time step of $\Delta t = 5\Delta x$.

Fig. 7.4 depicts the results of our numerical predictions for the vertical velocity of the particle and for the normalized height $\left(\frac{H-d_p}{2d_p}\right)$ with the parameters given in Table 5. Those results show good agreement with the experimental results of

Table 5
Parameters used in the sedimentation problem of Section 7.3.

Case name	$\rho_f\,(\mathrm{kg/m^3})$	$\mu_f\,(10^{-3}\,\mathrm{Ns/m^2})$	$u_\infty\,(\mathrm{m/s})$	Re_p
C1	970	373	.038	1.5
C2	965	212	.06	4.1
C3	962	113	.091	11.6
C4	960	58	.128	31.9

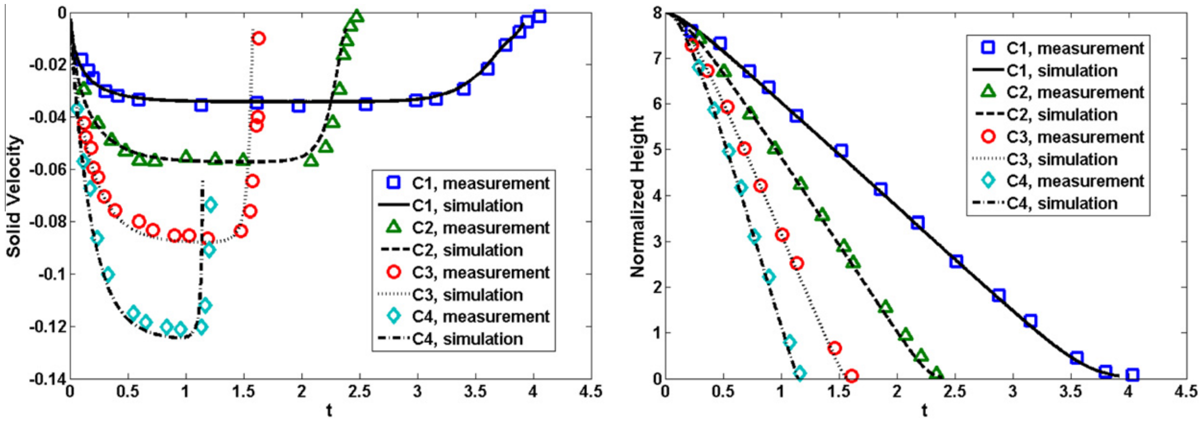


Fig. 7.4. Comparisons with the experimental data of Cate et al. [6] of the computed vertical velocity (left) and normalized height (right) in the case of the falling sphere example of Section 7.3. The symbols represent the experimental data while the solid lines depict the simulation results on a $120^2 \times 192$ grid.

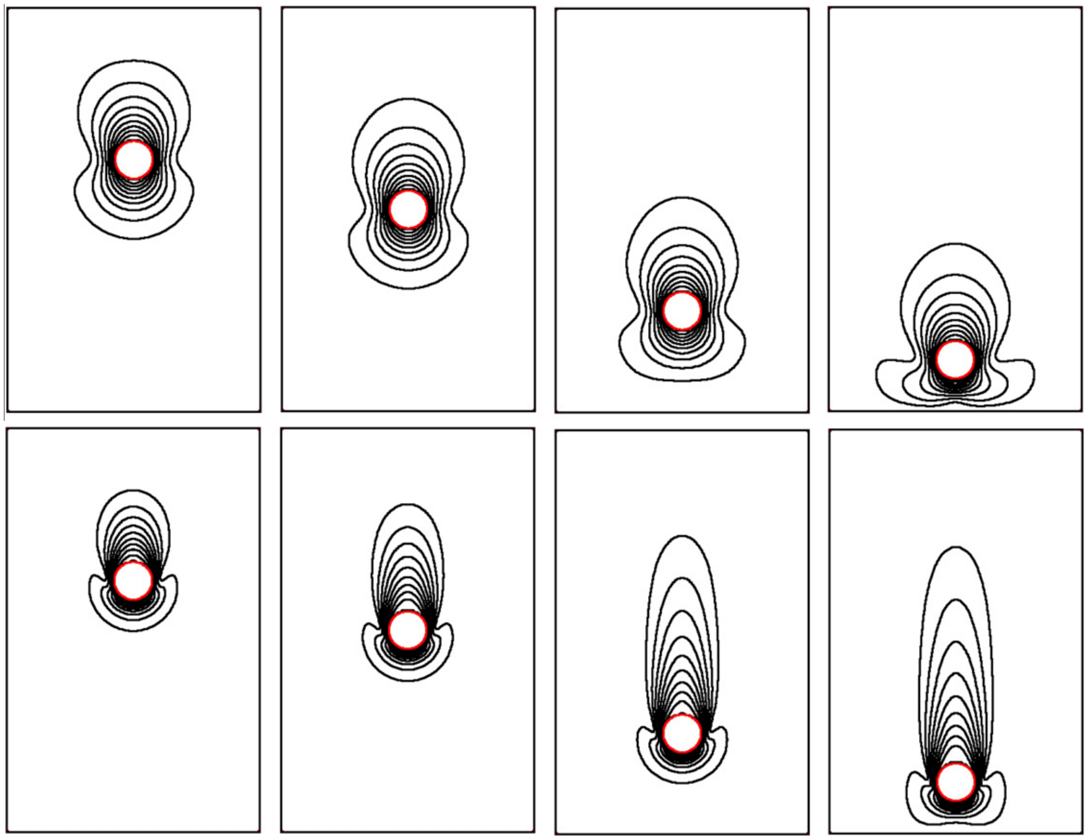


Fig. 7.5. Contours of the normalized velocity magnitude $\left(\frac{\|U\|}{U_\infty}\right)$ when the particle's center is at location.1, .08, .04, and.02, from left to right. The top figures correspond to the case where $Re = 1.5$, while the bottom figures correspond to the case where $Re = 31.9$. The contour values range between 0 and 1 with equal spacing of.1.

Table 6
Convergence study for the falling sphere example of Section 7.3. The convergence rate for the nth row is computed as $\log_{\left|\frac{\Delta x_{n-1}}{\Delta x_n}\right|}\left|\frac{v_{n-1}-v_{\text{ref}}}{v_n-v_{\text{ref}}}\right|$, where v_{ref} is the computed solution on a $140^2 \times 224$ grid.

grid	$\|v - v_{140^2\times 224}\|_\infty$	rate
$80^2 \times 120$	8.44×10^{-3}	
$90^2 \times 144$	5.98×10^{-3}	2.92
$100^2 \times 160$	3.73×10^{-3}	4.47
$110^2 \times 176$	2.49×10^{-3}	4.24
$120^2 \times 192$	1.75×10^{-3}	4.05

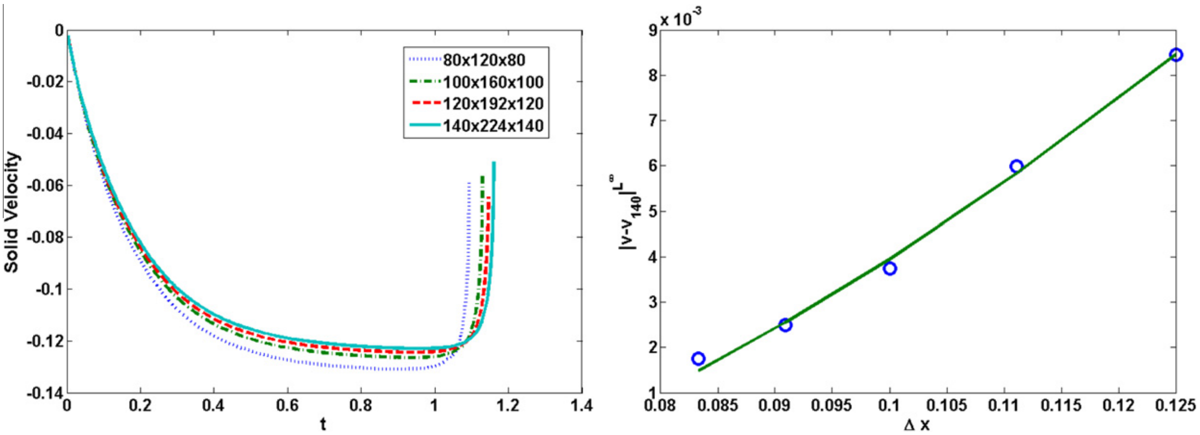


Fig. 7.6. Convergence study on particle setting velocity for the $Re = 31.9$ case. The left figure depicts the convergence under grid refinement. The right figure depicts the curve fitting of the error $e = C((\Delta x)^p - (\Delta\, x_{140})^p)$ with $C = 0.7577$ and $p = 1.97$. Here the order of accuracy is found to be 1.97. In each case, the error is measured over the time interval $t \in [0, 1]$ in the L^∞-norm.

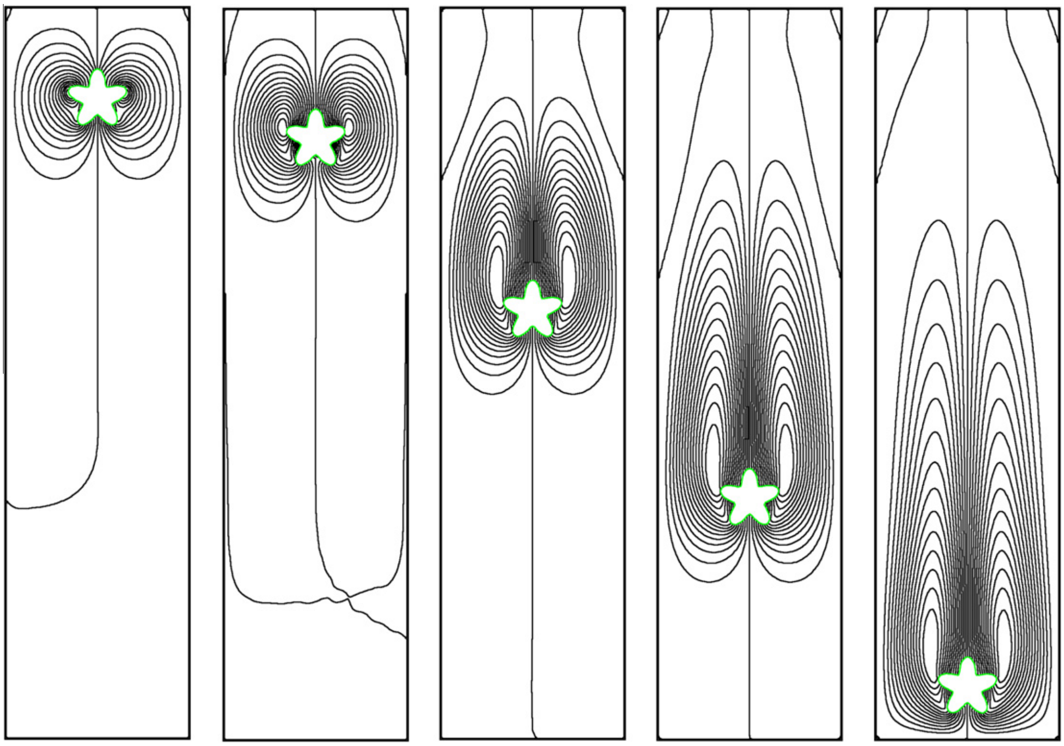


Fig. 7.7. Streamlines of a falling star-shaped object: the grid resolution is 80 × 320, and the time from left to right is $t = 0$, 0.1, 0.3, 0.5, and 0.7. The stream function ψ is calculated from solving $-\Delta\psi = \nabla \times \boldsymbol{U}$ with a homogeneous Dirichlet boundary condition on the boundary of the domain. For each time step, we have drawn 31 contour lines whose contour values are evenly sampled between the minimum and maximum values of the stream function.

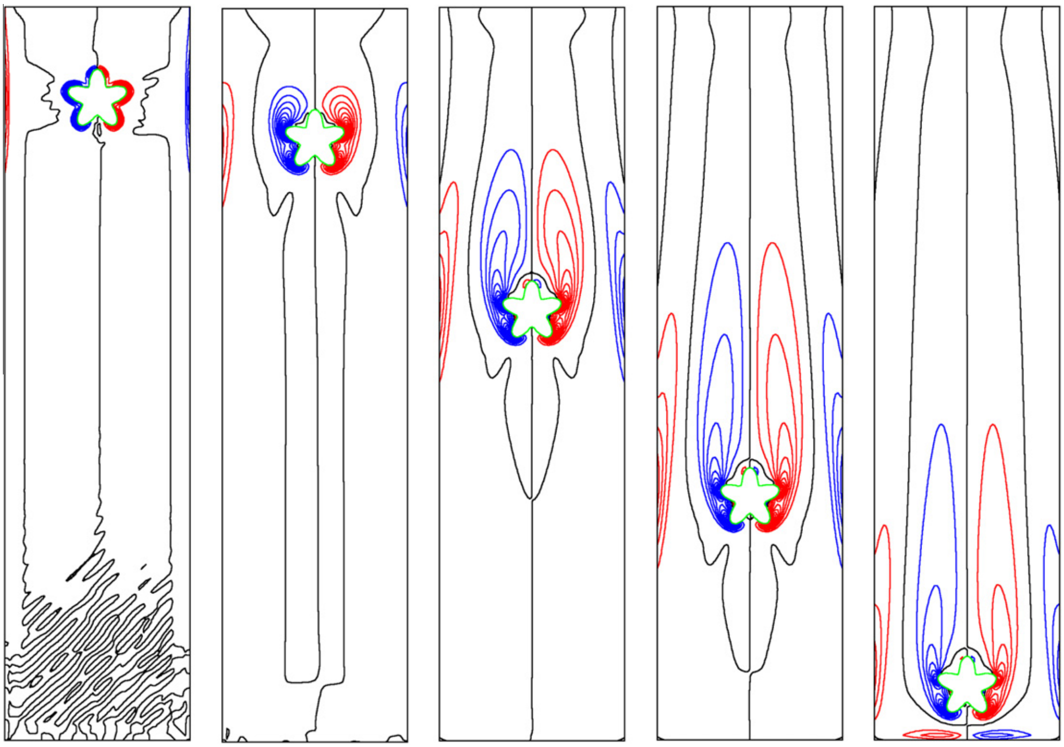


Fig. 7.8. Vorticity contours of a falling star-shaped object: the grid resolution is 80 × 320, and the time from left to right is $t = 0, 0.1, 0.3, 0.5$, and 0.7. For each time step, we have drawn 31 contour lines whose contour values are evenly sampled between the minimum and maximum values of the vorticity.

Ten Cate et al. [6]. Fig. 7.5 depicts the contour of the normalized velocity magnitude $\frac{\|u\|}{u_\infty}$. Those results are in agreement with those obtained with the simulations in Apte et al. [1].

For accuracy analysis, numerical solutions are calculated on grid resolutions of $80^2 \times 120$, $90^2 \times 144$, $100^2 \times 160$, $110^2 \times 176$, $120^2 \times 196$, and $140^2 \times 224$. The finest one among them is taken as the reference solution. Table 6 shows about fourth order convergence of the numerical solutions. The order is obviously overestimated since the reference solution is not computed with suitably fine resolution. Thus, we follow the analysis of Section 7.2 and use the data to fit the following error $e = C((\Delta x)^p - (\Delta x_{140})^p)$, where Δx_{140} represent the grid spacing when 140 grid points are used in the x-direction. We find $C = 0.7567$ and $p = 1.9667$ using the nonlinear least-square fit command `nlinfit` in MATLAB, as illustrated in the right part of Fig. 7.6. This analysis confirms that our numerical method is second-order accurate in the maximum norm.

7.4. Falling star-shaped object in 2D

In this example, we provide the results of a simulation of an arbitrary shaped rigid body falling in a liquid. The rigid body is described by the following level-set function:

$$\phi(x,y) = \frac{r}{0.005} - 1 - \frac{y^5 + 5x^4y - 10x^2y^3}{3r^5}$$

on a computational domain $[-0.2, 0.2] \times [-0.6, 0.2]$, and $r = \sqrt{x^2 + y^2}$. The fluid density is taken as $\rho_f = 1000\ \mathrm{kg\ m^{-2}}$, while the density of the rigid body is $\rho_s = 2000\ \mathrm{kg\ m^{-2}}$. The gravity constant is $g = 9.8$ and the viscosity of the fluid is $\mu = 0.05\ \mathrm{kg\ s^{-1}}$. Non-slip boundary conditions are imposed on the left, right, and bottom walls; the outflow boundary condition is imposed on the top wall. Figs. 7.7 and 7.8 depict the streamlines and the vorticity contours obtained on a 80 × 320 uniform grid. In this simulation, we use a time step of $\Delta t = 2\Delta x$.

8. Conclusion

We have introduced a novel numerical method to simulate the two-way fluid/solid coupling. Our approach is monolithic since it takes into account the full coupling between the fluid and the rigid body in a single step. It produces a linear system for the projection step that is symmetric positive definite and we have described how we can use the preconditioner of standard fluid simulations to efficiently solve the linear system in the case of fluid/solid coupling using a conjugate gradient method. The monolithic treatment results in a stable projection step, i.e. the kinetic energy does not increase in the projection step. Numerical results indicate that the method is second-order accurate in the L^∞-norm and agrees quantitatively with experimental results.

Acknowledgements

The research of F. Gibou was supported in part by ONR under grant agreement N00014–11-1–0027, by the National Science Foundation under grant agreement CHE 1027817, by the Department of Energy under grant agreement DE-FG02–08ER15991, by the Institute for Collaborative Biotechnologies through contract No. W911NF-09-D-0001 from the U.S. Army Research Office and by the W.M. Keck Foundation. The research of C. Min was supported in part by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010–0028298) and by the Korea Research Foundation Grant funded by the Korean Government (KRF-2011–0013649).

References

[1] S.V. Apte, M. Martin, N.A. Patankar, A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows, J. Comput. Phys. 228 (8) (2009) 2712–2738.
[2] T. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2004) 349–355.
[3] C. Batty, F. Bertails, R. Bridson, A fast variational framework for accurate solid-fluid coupling, ACM Trans. Graph. (SIGGRAPH Proc.) 26 (3) (2007).
[4] J.B. Bell, P. Colella, H.M. Glaz, A second order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys 85 (1989) 257–283.
[5] D. Brown, R. Cortez, M. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.
[6] A.T. Cate, J.J. Derksen, L.M. Portela, E.A. Harry, D.A. Van, Fully resolved simulations of colliding monodisperse spheres in forced isotropic turbulence, J. Fluid Mech. 519 (2004) 233–271.
[7] A. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[8] M. Coquerelle, G. Cottet, A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies, J. of Comput. Phys. 227 (21) (2008) 9121–9137.
[9] G.-H. Cottet, P. Koumoutsakos, Vortex Methods – Theory and Practice, Cambridge University Press, New York, 2000.
[10] C. Farhat, M. Lesoinne, P. Le Tallec, Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity, Comput. Method Appl. Mech. Eng. 157 (1-2) (1998) 95–114.
[11] Z. Feng, E. Michaelides, A direct forcing method in the simulations of particulate flows, J. Comput. Phys. 202 (1) (2005) 20–51.
[12] M. Garcia, J. Gutierrez, N. Rueda, Fluid–structure coupling using lattice–Boltzmann and fixed-grid FEM, Finite Elements in Analysis and Design 47 (8) (2011) 906–912.
[13] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2005) 577–601.
[14] F. Gibou, R. Fedkiw, L.-T. Cheng, M. Kang, A second-order–accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.
[15] G. Golub, C. Loan, Matrix Computations, The John Hopkins University Press, 1989.
[16] J.L. Guermond, J. Shen, On the error estimates for the rotational pressure-correction projection methods, Math. Comput. 73 (2003) 1719–1737.
[17] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluids with free surfaces, Phys. Fluids 8 (1965) 2182–2189.
[18] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[19] P. McCorquodale, P. Colella, D. Grote, J.-L. Vay, A node-centered local refinement algorithm for Poisson's equation in complex geometries, J. Comput. Phys. 201 (2004) 34–60.
[20] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equation on non-graded adaptive grids, J. Comput. Phys. 219 (2006) 912–929.
[21] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (2007) 300–321.
[22] C. Min, F. Gibou, Robust second order accurate discretizations of the multi-dimensional heaviside and dirac delta functions, J. Comput. Phys. 227 (2008) 9686–9695.
[23] Y. Ng, H. Chen, C. Min, F. Gibou, Guidelines for poisson solvers on irregular domains with dirichlet boundary conditions using the ghost fluid method, J. Sci. Comput. 41 (2009) 300–320.
[24] Y. Ng, C. Min, F. Gibou, An efficient fluid-solid coupling algorithm for single-phase flows, J. Comput. Phys. 228 (2009) 8807–8829.
[25] C. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[26] C. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.
[27] J.W. Purvis, J.E. Burkhalter, Prediction of critical mach number for store configurations, AIAA J. 17 (1979) 1170–1177.
[28] A. Robinson-Mosher, C. Schroeder, R. Fedkiw, A symmetric positive definite formulation for monolithic fluid structure interaction, J. Comput. Phys. 230 (2011) 1547–1566.
[29] A. Robinson-Mosher, T. Shinar, J. Gretarsson, J. Su, R. Fedkiw, Two-way coupling of fluids to rigid and deformable solids and shells, ACM Trans. Graph. 27 (46) (2008).
[30] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing, New York, NY, 1996.
[31] G.H. Shortley, R. Weller, Numerical solution of laplace's equation, J. Appl. Phys. 9 (1938) 334–348.
[32] P. Smereka, The numerical approximation of a delta function with application to level set methods, J. Comput. Phys. 211 (2006) 77–90.
[33] J. Towers, Finite difference methods for approximating heaviside functions, J. Comput. Phys. 228 (2009) 3478–3489.
[34] L. Trefethen, D. Bau III, Numerical Linear Algebra, SIAM, 1997.
[35] H. Wang, J. Chessa, W.K. Liu, T. Belytschko, The immersed/fictitious element method for fluid–structure interaction: volumetric consistency, compressibility and thin members, Int. J. Numer. Methods Eng. 74 (2008) 32–55.
[36] D. Xiu, G. Karniadakis, A semi-Lagrangian high-order method for Navier–Stokes equations, J. Comput. Phys 172 (2001) 658–684.