

The volume-filtering immersed boundary method

Himanshu Dave, Marcus Herrmann, M. Houssem Kasbaoui*

School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, 85281, AZ, USA

ARTICLE INFO

Article history:

Received 30 September 2022

Received in revised form 24 March 2023

Accepted 7 April 2023

Available online 27 April 2023

Keywords:

Immersed boundary method

Volume-filtering

Fully-resolved simulations

CFD

ABSTRACT

We present a novel framework to deal with static and moving immersed boundaries (IB) based on volume-filtering. In this strategy, called Volume-Filtering Immersed Boundary (VFIB) method, transport equations are derived by filtering the Navier-Stokes equations and accounting for stresses at the solid-fluid interface. The result is that boundary conditions that normally apply on the solid-fluid interface are transformed into bodyforces that apply on the right-hand side of the filtered transport equations. In this method, the filter width acts as a parameter that controls the level of resolution. The IB is considered well-resolved if the filter width is much smaller than the characteristic corrugation scale of the interface. There are several innovations in this IB method. First, it sheds light on the role of the internal flow which arises when the transport equations are solved inside the IB. We show that, it is essential to separate stresses due to the external and internal fluids in order to get accurate forces, and provide a method to do so. Second, we show that the volumes associated with Lagrangian forcing points on the boundary depend on the local topology of the surface. We provide a straightforward way to compute these volumes using a triangle tessellation of the interface and the surface density function. Third, we provide an efficient procedure to compute the solid volume fraction, thus, enabling tagging interior/exterior cells. This volume fraction is also involved in the procedure to separate stresses due to the external fluid from the total stresses. Fourth, we show a path forward to extend the VFIB method to Large Eddy Simulations involving IBs. Lastly, we apply the VFIB in several numerical tests involving two- and three-dimensional static and moving IBs. We show greatly improved results compared to prior IB methods. Further, we test several filter kernels and show that, for well-resolved IBs, the choice of the kernel plays little role.

© 2023 Elsevier Inc. All rights reserved.

1. Introduction

Besides few canonical flows, most fluid dynamic applications involve bounding surfaces with complex topology. These surfaces may be fixed, as in the case of airfoils and ship hulls, or moving, as in the case of stirred-tank reactors, flapping wings, and turbines. Despite increasing computing power, numerical simulations of such flows hinge on the availability of methods capable of capturing flow-surface interactions with accuracy while also meeting practical computational considerations such as robustness, ease of implementation, and scalability. The seminal work of Peskin [28,27,18,29], and later extended by several investigators, shows that it is possible to impose boundary conditions on topologically complex bounding surfaces without resorting to body-fitted meshes. This approach enables the use of fast and scalable Cartesian grid

* Corresponding author.

E-mail address: houssem.kasbaoui@asu.edu (M.H. Kasbaoui).

solvers. Boundary conditions are imposed using ad-hoc forcing terms added to the right-hand side of the governing equations. Conceptually, the equations to be solved for an incompressible flow are

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}_{\text{IB}} \quad (2)$$

where ρ and ν are the fluid's constant density and kinematic viscosity, respectively. The ad-hoc numerical term $\mathbf{F}_{\text{IB}}$ represents the immersed boundary (IB) forcing and is crafted to impose no-slip boundary conditions. Despite the popularity of IB methods, the ad-hocness of these methods remains a problem: the forcing term $\mathbf{F}_{\text{IB}}$ does not correspond to any physical term in the original Navier-Stokes equations. This makes it difficult to confidently answer long-standing questions such as (i) should the forcing be applied to the entire volumetric region occupied by the immersed object or be limited to the solid-fluid boundary? (ii) how is the solution affected by the sharpness of the immersed boundary representation?, and (iii) how to properly compute the hydrodynamic force on the solid when the immersed boundary is diffuse? Further, the fact that the IB forcing does not derive from analytical expressions makes it difficult to identify changes that would materially improve solution quality, extend the approach to Large-Eddy Simulations, or connect with established multiphase flow models in flows laden with a multitude of small immersed solids. In this manuscript, we remove the ad-hocness by deriving a new framework for immersed boundary methods that has sound theoretical footing. We show that the immersed boundary forcing can be derived rigorously by filtering the Navier-Stokes equations, discuss its discretization, and show that several other terms may be missing in previously proposed immersed boundary methods.

The work of Uhlmann [44] represents a notable improvement over Peskin's original method. Uhlmann [44] builds the immersed boundary forcing on a cloud of Lagrangian markers placed on the surface of the solid. Then, using convolutions with regularized Dirac delta [32], the Lagrangian forcing is transformed into the Eulerian forcing field $\mathbf{F}_{\text{IB}}$. The Lagrangian forcing is built such that, in principle, the interpolated fluid velocities at the Lagrangian markers match the solid's velocity at these locations. Uhlmann [44] shows that the method performs well in several benchmark tests with static and moving boundaries. Spurious oscillations of the hydrodynamic force observed with other immersed boundary methods [43,19,38,36] are significantly reduced. This makes Uhlmann's method remarkably stable even with thousands of fully resolved moving spheres. Since then, several improvements have been proposed. Yang et al. [47] introduced a smoothing technique for the discrete Dirac delta that further reduces spurious oscillations observed with moving immersed boundaries. Recognizing that the diffuse nature of the IB forcing causes the IB forcing from one Lagrangian marker to affect the calculation of IB forcing on neighboring markers, Luo et al. [23] proposed a variant, called multidirect forcing, where the IB forcing is imposed iteratively to improve the convergence of the Lagrangian marker velocity towards the desired no-slip velocity. Breugem [7] used this method in simulations with resolved spheres, and noticed that the Lagrangian markers must be retracted inwards to get correct hydrodynamic forces on the immersed spheres. Kempe and Fröhlich [17] proposed a variant similar to the multidirect forcing of Luo et al. [23] where the IB forcing is applied iteratively, and introduced a different approach for computing hydrodynamic forces on immersed solids based on level-set functions. Kasbaoui et al. [16] proposed a semi-implicit time integration scheme for the calculation of the IB forcing term based on an iterative Crank-Nicolson scheme. This approach improves the convergence of the Lagrangian marker velocity and was shown to compare well with experimental data even for inertially stirred turbulent flows in closed vessels.

Despite the success of the aforementioned methods, there are still open questions stemming from the ad-hocness of these methods. First, the role of the internal flow inside the immersed solid is not yet fully understood. This flow may develop when the IB forcing is applied only on the surface of the immersed solid, leaving internal cells unforced. With the so-called fictitious domain IB methods, a rigidity constraint is applied to enforce rigid body motion within the solid [12,39]. However, Uhlmann [44] reports that applying the forcing throughout the volumetric region does not change his results significantly compared to when the forcing is applied on the boundary only. Thus, the latter option is preferred due to its lower computational cost. Despite being considered an artificial byproduct of the forcing technique [13,44,17], the internal flow is used to compute hydrodynamic forces on moving spherical particles [44,17,42]. Uhlmann [44] assimilates the flow inside the sphere to rigid body motion despite it not being the case. Kempe and Fröhlich [17] show that it is necessary to embrace the non-rigid motion inside the sphere in order to compute the hydrodynamic force accurately. If the internal flow was to be neglected or zeroed out on the basis that it is artificial, then the hydrodynamic force computed by Kempe and Fröhlich [17] and Uhlmann [44] would vanish.

Another ambiguous point in this class of IB methods concerns the determination of Lagrangian marker volumes which arise in the computation of the Lagrangian immersed boundary forcing. Uhlmann [44] and several others (e.g. [17,42]) relate the marker volume ΔV_m to the grid spacing Δx following $\Delta V_m \sim \Delta x^3$, and use enough markers to form a thin shell around the immersed boundary. However, Zhou and Balachandar [49] argue that it is a misconception to pin ΔV_m to Δx^3 and suggest that it should be considered as a relaxation factor that controls how fast errors in the no-slip boundary conditions decay. They provide an expression to compute ΔV_m based on stability analysis, but, the calculation is cumbersome and only applicable to spherical immersed boundaries. In both [44] and [49], geometric information about the immersed boundary such as surface area and curvature are not taken into consideration. However, the fact that Zhou and Balachandar [49] find that ΔV_m must decrease with decreasing marker spacing suggests that the marker volume depends, at least in part, on the portion of the IB surface area around the marker.

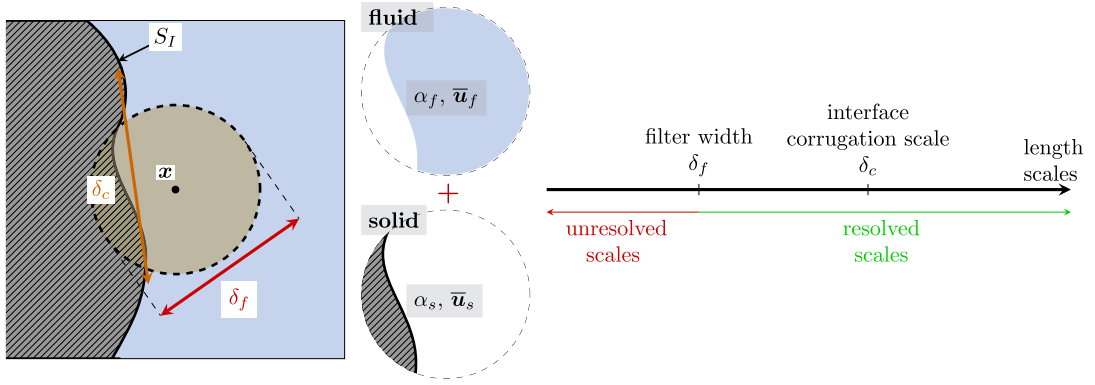


Fig. 1. Illustration of the volume-filtering approach. Filtering the point-wise fields allows the extraction of average fluid and solid volume fractions (α_f and α_s), mass ($\alpha_f \rho_f$ and $\alpha_s \rho_s$) and momentum ($\alpha_f \rho_f \bar{\mathbf{u}}_f$ and $\alpha_s \rho_s \bar{\mathbf{u}}_s$) within a region of size δ_f . The immersed boundary is well resolved when the characteristic corrugation scale δ_c of the interface is much larger than the filter width δ_f .

To overcome the issues raised above, we derive a new framework for immersed boundary methods that does not rely on any ad-hoc elements. The method we present, called Volume-filtering Immersed Boundary (VFIB) method, relies on the volume-filtering technique introduced by Anderson and Jackson [2]. The method is similar in spirit to Large Eddy Simulations, where the Navier-Stokes equations are filtered with a filter kernel having width δ_f . This procedure yields filtered governing equations where the effects of boundary conditions appear as right-hand side terms involving surface integrals on the immersed boundary. The derivation of these equations is physically and mathematically rigorous, and does not depend on any numerical considerations. Traditionally, volume-filtering has been applied to derive continuum equations for multiphase flows [2,3,14] and porous media [46]. In these applications, the width of the filter kernel, δ_f , is chosen much larger than the characteristic interface corrugation scale δ_c , such as particle radius or pore size. In this way, Eulerian quantities representing solid volume fractions, mass, and momentum can be extracted from large ensembles of discrete solids. For the purpose of deriving an immersed boundary method, we take the opposite limit: the filter width is much smaller than the characteristic interface corrugation scales. In this limit ($\delta_f \ll \delta_c$), the immersed boundary as well as all flow scales larger than δ_f is fully resolved. The resulting equations form the basis of the VFIB method. As we show in this manuscript, the VFIB method generalizes Uhlmann's immersed boundary method and sheds much needed clarification on the role of the internal flow, Lagrangian marker volume, and a more accurate way to compute hydrodynamic forces.

The objectives of the present manuscript are two-fold. The first goal is to present the theory underpinning the Volume-filtering Immersed Boundary method. This is done in section 2, where we present the derivation of the governing equations that are solved in the VFIB method. The emphasis here is on the mathematical and physical framework which does not depend on the choice of numerical parameters. With the theory clearly established, the second goal of this manuscript is to provide an implementation of the VFIB method, discuss practical considerations (such as the choice of filter kernel, relative size of filter kernel and mesh spacing, calculation of volume fraction, and forces on the IB), and demonstrate the approach in canonical test cases. We address the discretization and numerical implementation of these equations in section 3. In section 4, we discuss the characteristics of different filter kernels tested in the present study. We illustrate the approach using numerical examples with static, forcibly, and freely moving immersed boundaries in section 5. Finally, we give concluding remarks in section 6.

2. The volume-filtering immersed boundary method

In this section, we apply the volume-filtering method of Anderson and Jackson [2] to the problem of a fluid with an immersed solid. We consider two formulations and discuss the merits of both. In the *two-phase* formulation, we describe the dynamics of the flow outside the immersed boundary separately from the dynamics inside the solid. In the *one-phase* formulation, we consider that the immersed object is hollow and filled with the same fluid as outside of it. By describing the total mass and momentum conservation, we arrive at governing equations that can be discretized efficiently and where the Immersed Boundary forcing is given explicitly.

2.1. Two-phase formulation

Consider two-phases separated by an interface S_I as shown in the schematic in Fig. 1. Phase “f” corresponds to the fluid phase, while phase “s” corresponds to the immersed solid. Where the fluid exists, mass and momentum conservation are given by the incompressible Navier-Stokes equations

$$\nabla \cdot \mathbf{u}_f = 0 \quad (3)$$

$$\rho_f \left(\frac{\partial \mathbf{u}_f}{\partial t} + \nabla \cdot (\mathbf{u}_f \mathbf{u}_f) \right) = \nabla \cdot \boldsymbol{\tau}_f \quad (4)$$

where $\mathbf{u}_f$, p , and $\boldsymbol{\tau}_f = -p\mathbf{I} + \mu_f(\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T)$ are the fluid velocity, pressure, and stress tensor, respectively. Throughout, the manuscript we assume that the fluid density ρ_f and viscosity μ_f are constant. Borrowing the terminology from Jackson [14] equations (3) and (4) are called the *point-wise* conservation equations.

For the familiar Navier-Stokes equations (3) and (4), the boundary conditions are imposed through additional constraints that apply on the fluid-solid interface S_I and outside boundaries. Considering an impermeable interface, no slip-boundary conditions apply, i.e.,

$$\mathbf{u}_f(\mathbf{x}) = \mathbf{u}_I(\mathbf{x}) \quad \text{for } \mathbf{x} \in S_I \quad (5)$$

where $\mathbf{u}_I(\mathbf{x})$ is the velocity of a point $\mathbf{x}$ located on the interface.

To illustrate how volume-filtering works, consider a point $\mathbf{x}$ near the immersed boundary as schematized in Fig. 1. By filtering the point-wise quantities, we can extract the fluid volume fraction α_f , average fluid mass $\alpha_f \rho_f$, and fluid momentum $\alpha_f \rho_f \bar{\mathbf{u}}_f$ in a region of size δ_f around the point $\mathbf{x}$. In this way, we can define the filtered fluid velocity $\bar{\mathbf{u}}_f$ at location $\mathbf{x}$ corresponding to the average fluid velocity under the filter kernel. As shown in Fig. 1, it is only the volumetric region occupied by the fluid that counts towards the extracted fluid quantities. Likewise, we can extract the solid volume fraction α_s , average solid mass $\alpha_s \rho_s$, and solid momentum $\alpha_s \rho_s \bar{\mathbf{u}}_s$ around the point $\mathbf{x}$ by considering only the region occupied by the solid that falls under the filter kernel. However, because we do not seek to model the internal solid dynamics, we will focus on the fluid phase only. Contrary to the point-wise velocity $\mathbf{u}_f$, which exists only in the fluid region, the filtered fluid velocity $\bar{\mathbf{u}}_f$ can be defined for any arbitrary point in space. Moving the probing point $\mathbf{x}$ towards the solid causes the filtered fluid velocity $\bar{\mathbf{u}}_f$ to decay smoothly to zero as the fluid region under the filter kernel shrinks. Note that the filtered fluid velocity at points $\mathbf{x}$ inside the solid, but less than $\delta_f/2$ away from S_I , may not be zero as there is still a fluid region under the filter kernel. Thus, volume-filtering smears the effect of the interface over a region of size δ_f . The immersed boundary is well resolved when the filter width δ_f is much smaller than the characteristic corrugation scale δ_c of the immersed boundary. For example, if the immersed solid is a sphere of diameter D , we would require that $\delta_f \ll \delta_c = D$ to retain good resolution of the immersed boundary. Length scales below δ_f are unresolved and must be modeled. In the limit where δ_f is vanishingly small, the filtered velocity becomes a discontinuous field equal to the point-wise velocity inside the fluid and zero inside the solid.

To formalize this idea, we consider a *symmetric* filter kernel g that integrates to *unity* and has *compact* support of width δ_f . Mathematically, these properties read,

$$\iiint_{\mathbf{y} \in \mathbb{R}^3} g(\mathbf{y}) dV = 1, \quad (\text{unitary}) \quad (6)$$

$$g(-\mathbf{y}) = g(\mathbf{y}), \quad (\text{symmetric}) \quad (7)$$

$$g(\mathbf{y}) = 0 \text{ if } \|\mathbf{y}\| \geq \delta_f/2. \quad (\text{compact}) \quad (8)$$

Notice that the integration is considered over the entire space which includes regions occupied by both fluid and solid phases. Symmetry of the kernel is required to avoid artificial anisotropy and plays an important role in the derivation of the filtered conservation equations. Compactness of the filter kernel serves primarily a computational purpose as it allows fast numerical integration of g on surfaces, but is otherwise not required for the purpose of the derivation.

The fluid volume fraction at any arbitrary location $\mathbf{x}$ is given by

$$\alpha_f(\mathbf{x}, t) = \iiint_{\mathbf{y} \in \mathbb{R}^3} \mathbb{1}_f(\mathbf{y}, t) g(\mathbf{x} - \mathbf{y}) dV, \quad (9)$$

where $\mathbb{1}_f(\mathbf{y}, t)$ is an indicator function equal to 1 if $\mathbf{y}$ is in the fluid and 0 otherwise. Similarly, the solid volume fraction α_s is defined by replacing the fluid indicator function in equation (9), with that of the solid, i.e., $\mathbb{1}_s = 1 - \mathbb{1}_f$. The volume fraction $\alpha_f(\mathbf{x})$ represents the ratio of the volume occupied by the fluid to the total volume under the support of the filter kernel centered at a location $\mathbf{x}$. Regions of space occupied exclusively by the fluid have a fluid volume fraction $\alpha_f = 1$. Conversely, regions where only the solid phase can be found within the support of the filter kernel have a fluid volume fraction $\alpha_f = 0$. Locations where $0 < \alpha_f(\mathbf{x}) < 1$ correspond to those where both phases are within reach of the filter kernel, as in the schematic in Fig. 1. This corresponds to a narrow band of width δ_f around the interface. Isocontours $\alpha_f = \alpha_s = 0.5$ give the location of the interface, if the latter can be considered locally planar. In this formulation, discontinuous effects across the interface are smoothed over a distance δ_f . The interface representation can be made sharper by reducing the size of δ_f . Equation (6) guarantees that the solid and fluid volume fractions add up to unity at any given location, i.e., $\alpha_f(\mathbf{x}) + \alpha_s(\mathbf{x}) = 1$ for any $\mathbf{x} \in \mathbb{R}^3$. At this point, we emphasize that the volume fraction in equation (9), filtered velocities, and filtered governing equations to follow are not tied to any specific discretization or mesh.

For an arbitrary point $\mathbf{x}$ in space, the volume-filtered fluid velocity $\bar{\mathbf{u}}_f$ is defined as following,

$$\alpha_f(\mathbf{x}, t) \bar{\mathbf{u}}_f(\mathbf{x}, t) = \iiint_{\mathbf{y} \in \mathbb{R}^3} \mathbb{1}_f(\mathbf{y}, t) \mathbf{u}_f(\mathbf{y}, t) g(\mathbf{x} - \mathbf{y}) dV. \quad (10)$$

The volume-filtered velocity is *continuous* and *exists everywhere*. It tends smoothly to 0 a distance $\delta_f/2$ away within the solid.

The volume-filtered governing equations are derived by application of the filter to the point-wise equations (3) and (4). Using the divergence theorem and symmetry of the filter kernel ($\partial g(\mathbf{x} - \mathbf{y})/\partial x_i = -\partial g(\mathbf{x} - \mathbf{y})/\partial y_i$), one can show that filtering the gradient, divergence and time derivative operators yields,

$$\alpha_f(\mathbf{x}) \overline{\nabla \Lambda}(\mathbf{x}) = \nabla(\alpha_f \bar{\Lambda}) - \iint_{\mathbf{y} \in S_I} \mathbf{n} \Lambda(\mathbf{y}, t) g(\mathbf{x} - \mathbf{y}) dS, \quad (11)$$

$$\alpha_f(\mathbf{x}) \overline{\nabla \cdot \Lambda}(\mathbf{x}) = \nabla \cdot (\alpha_f \bar{\Lambda}) - \iint_{\mathbf{y} \in S_I} \mathbf{n} \cdot \Lambda(\mathbf{y}, t) g(\mathbf{x} - \mathbf{y}) dS, \quad (12)$$

$$\alpha_f(\mathbf{x}) \overline{\frac{\partial \Lambda}{\partial t}}(\mathbf{x}) = \frac{\partial(\alpha_f \bar{\Lambda})}{\partial t} + \iint_{\mathbf{y} \in S_I} (\mathbf{n} \cdot \mathbf{u}) \Lambda(\mathbf{y}, t) g(\mathbf{x} - \mathbf{y}) dS, \quad (13)$$

where Λ is an arbitrary vector or tensor property of the fluid. Here, $\mathbf{n}$ represents the normal vector at the interface pointing from the solid to the fluid phase. A full derivation of these identities can be found in the original work of Anderson and Jackson [2]. Because the application of the filter removes the notion of a boundary ($\mathbf{x}$ can be anywhere in $\mathbb{R}^3$), information from the boundary conditions emerges in identities (11), (12), and (13) as surface integrals on the interface separating the solid and fluid phases.

Applying the filtering procedure to the mass and momentum equations (3) and (4) leads to

$$\frac{\partial \alpha_f}{\partial t} + \nabla \cdot (\alpha_f \bar{\mathbf{u}}_f) = 0, \quad (14)$$

$$\rho_f \left(\frac{\partial}{\partial t} (\alpha_f \bar{\mathbf{u}}_f) + \nabla \cdot (\alpha_f \bar{\mathbf{u}}_f \bar{\mathbf{u}}_f) \right) = \nabla \cdot (\alpha_f \bar{\boldsymbol{\tau}}_f) - \iint_{\mathbf{y} \in S_I} \mathbf{n} \cdot \boldsymbol{\tau}_f(\mathbf{y}, t) g(\mathbf{x} - \mathbf{y}) dS. \quad (15)$$

In the filtered momentum equation (15), the term

$$\mathbf{F}_{I,f}(\mathbf{x}, t) = \iint_{\mathbf{y} \in S_I} \mathbf{n} \cdot \boldsymbol{\tau}_f(\mathbf{y}, t) g(\mathbf{x} - \mathbf{y}) dS \quad (16)$$

represents a force density exerted by the immersed solid on the fluid. Note that this term includes a surface integral on the immersed boundary, meaning that the forcing is limited to a thin region of width δ_f around the immersed boundary rather than the entire volumetric region. The force exerted by the fluid on the solid can be obtained by integrating the immersed boundary force density over the entire domain (fluid and solid regions):

$$\iint_{\mathbf{y} \in S_I} \mathbf{n} \cdot \boldsymbol{\tau}_f(\mathbf{y}, t) dS = \iiint_{\mathbf{x} \in \mathbb{R}^3} \mathbf{F}_{I,f}(\mathbf{x}, t) dV. \quad (17)$$

The relationship above does not require $\delta_f = 0$, and holds true for arbitrary filter widths and surface curvatures. The filtered stress tensor $\alpha_f \bar{\boldsymbol{\tau}}_f = \alpha_f [-p\mathbf{I} + \mu_f (\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T)]$ requires detailed examination. Whereas filtering the pressure part is straightforward ($-\alpha_f \bar{p}\mathbf{I} = -\alpha_f \bar{p}\mathbf{I}$), filtering the viscous part leads to

$$\mu_f \alpha_f \left(\overline{\nabla \mathbf{u}_f} + \overline{\nabla \mathbf{u}_f^T} \right) = \mu_f \alpha_f \left(\nabla \bar{\mathbf{u}}_f + \nabla \bar{\mathbf{u}}_f^T - \frac{2}{3} (\nabla \cdot \bar{\mathbf{u}}_f) \mathbf{I} \right) + \alpha_f \mathbf{R}_{\mu,f} \quad (18)$$

where $\mathbf{R}_{\mu,f}$ represents the residual viscous stress tensor. Equation (18) is derived by application of the identity (11). The residual viscous stress tensor $\mathbf{R}_{\mu,f}$ is zero away from the immersed boundary. Near the IB, the effects of $\mathbf{R}_{\mu,f}$ may be significant if the IB is poorly resolved. In the limit where $\delta_f \rightarrow 0$, the residual viscous stresses vanish.

Application of the filter leads to the emergence of unclosed convective terms $\alpha_f \bar{\mathbf{u}}_f \bar{\mathbf{u}}_f$ in equation (15). To deal with these terms, we introduce the subfilter-scale stress tensor,

$$\boldsymbol{\tau}_{\text{sfs},f} = \bar{\mathbf{u}}_f \bar{\mathbf{u}}_f - \bar{\mathbf{u}}_f \bar{\mathbf{u}}_f. \quad (19)$$

To summarize, the set of equations obtained by volume-filtering equations (3) and (4) are:

$$\frac{\partial \alpha_f}{\partial t} + \nabla \cdot (\alpha_f \bar{\mathbf{u}}_f) = 0, \quad (20)$$

$$\rho_f \left(\frac{\partial}{\partial t} (\alpha_f \bar{\mathbf{u}}_f) + \nabla \cdot (\alpha_f \bar{\mathbf{u}}_f \bar{\mathbf{u}}_f) \right) = \nabla \cdot \left(-\alpha_f \bar{p} \mathbf{I} + \alpha_f \mu_f \left(\nabla \bar{\mathbf{u}}_f + \nabla \bar{\mathbf{u}}_f^T - \frac{2}{3} (\nabla \cdot \bar{\mathbf{u}}_f) \mathbf{I} \right) + \alpha_f \mathbf{R}_{\mu,f} \right) - \mathbf{F}_{l,f} - \nabla \cdot (\alpha_f \boldsymbol{\tau}_{\text{sfs},f}). \quad (21)$$

While the derivation of equations (20) and (21) is mathematically rigorous, discretizing and solving these equations in their present form is challenging. The main complication is due to the fluid volume fraction that vanishes inside the solid. Because of this, the filtered fluid velocity $\bar{\mathbf{u}}_f$ cannot be computed from the transported quantity $(\alpha_f \bar{\mathbf{u}}_f)$ without leading to large errors and computational instabilities.

Lastly, as we discuss in Appendix A, equations (20) and (21) can be further transformed to arrive at a form that is frequently used in multiphase flows, see equation (A.5). However, such procedure is not needed for the present purpose.

2.2. One-phase formulation

We now present an alternative formulation that overcomes the numerical stability issues encountered with the two-phase formulation. In the one-phase formulation, we assume that the immersed solid is hollow and filled with fluid having identical density and viscosity as the fluid outside. In this view, the immersed boundary represents an infinitely thin interface that separates two fluids. We denote α_1 and α_2 the volume fractions occupied by the exterior and interior fluids, respectively. Applying volume-filtering to both fluids, we obtain the following mass and momentum conservation equations,

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \bar{\mathbf{u}}_i) = 0, \quad (22)$$

$$\rho_f \left(\frac{\partial (\alpha_i \bar{\mathbf{u}}_i)}{\partial t} + \nabla \cdot (\alpha_i \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i) \right) = \nabla \cdot \left(-\alpha_i \bar{p}_i \mathbf{I} + \alpha_i \mu_f \left(\nabla \bar{\mathbf{u}}_i + \nabla \bar{\mathbf{u}}_i^T - \frac{2}{3} (\nabla \cdot \bar{\mathbf{u}}_i) \mathbf{I} \right) + \alpha_i \mathbf{R}_{\mu,i} \right) - \mathbf{F}_{l,i} - \nabla \cdot (\alpha_i \boldsymbol{\tau}_{\text{sfs},i}), \quad (23)$$

where $i = 1$ or 2 depending on which fluid is considered. In addition to the above equations, the volume fractions are constrained by the condition $\alpha_1 + \alpha_2 = 1$.

To obtain single-field equations, we sum over the two fluids in equations (22) and (23):

$$\nabla \cdot \mathbf{u}_m = 0, \quad (24)$$

$$\rho_f \left(\frac{\partial \mathbf{u}_m}{\partial t} + \nabla \cdot (\mathbf{u}_m \mathbf{u}_m) \right) = -\nabla p_m + \mu_f \nabla^2 \mathbf{u}_m - \mathbf{F}_{l,m} + \nabla \cdot (\boldsymbol{\tau}_r - \boldsymbol{\tau}_{\text{sfs},m}). \quad (25)$$

These equations describe the transport of the total mass and momentum of both fluids. Here, $\mathbf{u}_m = \alpha_1 \bar{\mathbf{u}}_1 + \alpha_2 \bar{\mathbf{u}}_2$ is the mixture velocity, $p_m = \alpha_1 \bar{p}_1 + \alpha_2 \bar{p}_2$ is the mixture pressure, $\mathbf{F}_{l,m} = \mathbf{F}_{l,1} + \mathbf{F}_{l,2}$, and $\boldsymbol{\tau}_{\text{sfs},m} = \boldsymbol{\tau}_{\text{sfs},1} + \boldsymbol{\tau}_{\text{sfs},2}$. The viscous stress tensors and residual stress tensors $\mathbf{R}_{\mu,i}$ combine to give the term $\mu_f \nabla^2 \mathbf{u}_m$ in equation (25). For ease of notation, we drop the subscript m in the rest of the manuscript.

In this one-phase formulation, we see the emergence of a new tensor,

$$\boldsymbol{\tau}_r = \mathbf{u} \mathbf{u} - \sum_i \alpha_i \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i = \alpha_1 \alpha_2 (\bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_2)(\bar{\mathbf{u}}_2 - \bar{\mathbf{u}}_1). \quad (26)$$

The tensor $\boldsymbol{\tau}_r$ represents the momentum drift across the interface. This tensor can be neglected for sufficiently well resolved immersed boundaries, i.e., $\delta_f / \delta_c \ll 1$. Away from the interface, $\boldsymbol{\tau}_r = 0$ since $\alpha_1 = 0$ or $\alpha_2 = 0$. Near the interface, no-slip boundary conditions lead to $\bar{\mathbf{u}}_1 \simeq \bar{\mathbf{u}}_2 \simeq \mathbf{u}_I$, where $\mathbf{u}_I$ is the interface velocity, hence, $\boldsymbol{\tau}_r \simeq 0$. In the rest of the paper, we will assume that the immersed boundary representation is sufficiently sharp such that $\boldsymbol{\tau}_r$ can be considered identically zero everywhere. This assumption is equivalent to stating $\alpha_1 \alpha_2 (\bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_2) = 0$, and allows us to extract the internal and external fluid velocities using:

$$\alpha_1 \bar{\mathbf{u}}_1 = \alpha_1 \mathbf{u} + \alpha_1 \alpha_2 (\bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_2) \simeq \alpha_1 \mathbf{u} \quad (27)$$

$$\alpha_2 \bar{\mathbf{u}}_2 = \alpha_2 \mathbf{u} + \alpha_1 \alpha_2 (\bar{\mathbf{u}}_2 - \bar{\mathbf{u}}_1) \simeq \alpha_2 \mathbf{u} \quad (28)$$

In the one-phase formulation, the subfilter scale term is

$$\boldsymbol{\tau}_{\text{sfs}} = \sum_i \alpha_i \boldsymbol{\tau}_{\text{sfs},i} = \alpha_1 (\overline{\mathbf{u}_1 \mathbf{u}_1} - \bar{\mathbf{u}}_1 \bar{\mathbf{u}}_1) + \alpha_2 (\overline{\mathbf{u}_2 \mathbf{u}_2} - \bar{\mathbf{u}}_2 \bar{\mathbf{u}}_2) \quad (29)$$

In general, the subfilter-scale tensor must be closed if the immersed boundary is poorly resolved ($\delta_f/\delta_c = O(1)$ or $\delta_f/\delta_c \gg 1$). In particle-laden flows, where similar volume-filtering is carried out using filter kernels typically much larger than the particle diameters, the subfilter scale stresses are known as pseudo-turbulent stresses and are subject of active research and modeling [37,25]. Closures may also be required if the flow scales δ_u are not well-resolved ($\delta_f/\delta_u \gtrsim 1$), even if the immersed boundary is well-resolved $\delta_f/\delta_c \ll 1$. For example, if two well-resolved immersed boundaries approach one another, but the gap between the two solids becomes smaller than the filter width δ_f , τ_{sfs} should be augmented with a lubrication model [9] to represent the effect of the subfilter fluid in the gap between the two objects. In the case where the flow is turbulent and the filter width is larger than the turbulence scales, closure of τ_{sfs} , away from the immersed boundary, may be carried out by any of the well-established LES models such as turbulent eddy viscosity model of Boussinesq [6], the dynamic Smagorinsky model [11,20], or scale-similarity models [5]. While there is an array of closure models available to model τ_{sfs} , this paper focuses on the validation of the VFIB method in the limit of well resolved immersed boundaries ($\delta_f/\delta_c \ll 1$) and flow scales ($\delta_f/\delta_u \ll 1$), and therefore the subfilter-scale terms will be neglected.

The total immersed boundary force density can be written as

$$\mathbf{F}_I = \mathbf{F}_{I,1} + \mathbf{F}_{I,2} = \iint_{S_I} (\mathbf{n} \cdot [\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2]) (\mathbf{y}) g(\mathbf{x} - \mathbf{y}) dS \quad (30)$$

where $\mathbf{n}$ is the normal pointing from the internal fluid ($i = 2$) to the external fluid ($i = 1$). It is important to acknowledge that this term accounts for forces exerted on the immersed boundary by both fluids inside and outside. Computing the hydrodynamic force on an immersed object requires extracting the contribution of the external fluid, $\mathbf{F}_{I,1}$, from the total immersed boundary force density $\mathbf{F}_I$. This aspect is addressed in §2.4.

The force density $\mathbf{F}_I$ can be expressed explicitly using mixture quantities. To do so, consider the Taylor series of the total surface stresses in the vicinity of a point $\mathbf{y}_I$ located at the interface

$$(\mathbf{n} \cdot [\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2]) (\mathbf{y}) = (\mathbf{n} \cdot [\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2]) (\mathbf{y}_I) + (\mathbf{y} - \mathbf{y}_I) \cdot \nabla (\mathbf{n} \cdot [\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2]) (\mathbf{y}_I) + O(\delta_f^2) \quad (31)$$

Introducing equation (31) in equation (25) and rearranging the terms leads to

$$(\mathbf{n} \cdot [\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2]) (\mathbf{y}_I) = -\ell(\mathbf{y}_I) \left(\rho_f \frac{D\mathbf{u}}{Dt} - \nabla \cdot (-p\mathbf{I} + \mu_f(\nabla\mathbf{u} + \nabla\mathbf{u}^T) + \boldsymbol{\tau}_r - \boldsymbol{\tau}_{\text{sfs}}) \right) \Big|_{\mathbf{y}=\mathbf{y}_I} + O(\delta_f^2) \quad (32)$$

where D/Dt is the substantial derivative. Thanks to the symmetry of the filter kernel g , the first order term from the Taylor series cancels out leading to a second order approximation in δ_f of the immersed boundary stresses.

In equation (32), the length

$$\ell(\mathbf{y}_I) = \left(\iint_{S_I} g(\mathbf{y}_I - \mathbf{y}) dS \right)^{-1} \quad (33)$$

represents the interface smearing length at a location $\mathbf{y}_I$ on the immersed boundary. The smearing length is on the same order as the filter kernel width δ_f , however, the exact value depends on both interface curvature and filter kernel g . If the interface around point $\mathbf{y}$ can be considered flat, then $\ell(\mathbf{y}) = g(\mathbf{0})^{-1/3}$. For the triangle and cosine filter kernels described in §4, the smearing length for a flat interface is $\ell(\mathbf{y}) = 2/\delta_f$ and $\pi/(2\delta_f)$, respectively.

The smearing length is closely connected to the interface surface density. For a point $\mathbf{y}$ on the interface, equation (33) shows that $\ell(\mathbf{y})$ is the inverse of the surface density Σ at that location, where

$$\Sigma(\mathbf{x}) = \iint_{S_I} g(\mathbf{x} - \mathbf{y}) dS. \quad (34)$$

The surface density Σ represents how much surface area of the immersed boundary is under the filter kernel. Away from the interface, $\Sigma = 0$. Near the interface, the value of Σ depends on the choice of filter kernel, width δ_f , and local curvature. Integrating Σ over the entire domain gives the total surface area of the immersed boundary A_I :

$$\iiint_{\mathbf{x} \in \mathbb{R}^3} \Sigma(\mathbf{x}) dV = A_I. \quad (35)$$

Note that reducing δ_f increases Σ and, consequently, reduces the smearing length ℓ computed at a point on the interface.

At this point, the no-slip boundary condition can be introduced in equation (32). To do so, notice that a fluid particle located on the interface has an acceleration that matches the acceleration of the IB,

$$\frac{D}{Dt} \mathbf{u}(\mathbf{y}) = \frac{d}{dt} \mathbf{u}_I(\mathbf{y}) \quad \mathbf{y} \in S_I. \quad (36)$$

Once expressions (32) and (36) are introduced in (30), we see that the interface stress is given by:

$$\mathbf{F}_I(\mathbf{x}) = - \iint_{S_I} \ell(\mathbf{y}) \left(\rho_f \frac{d\mathbf{u}_I}{dt} + \nabla p - \mu_f \nabla^2 \mathbf{u} - \nabla \cdot (\boldsymbol{\tau}_r - \boldsymbol{\tau}_{\text{sfs}}) \right) \bigg|_{\mathbf{y}} g(\mathbf{x} - \mathbf{y}) dS + O(\delta_f^2) \quad (37)$$

Thus, the complete equations that must be solved in the one-phase VFIB method are

$$\nabla \cdot \mathbf{u} = 0, \quad (38)$$

$$\rho_f \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{u}) \right) = -\nabla p + \mu_f \nabla^2 \mathbf{u} + \nabla \cdot (\boldsymbol{\tau}_r - \boldsymbol{\tau}_{\text{sfs}}) + \iint_{S_I} \ell \left(\rho_f \frac{d\mathbf{u}_I}{dt} + \nabla p - \mu_f \nabla^2 \mathbf{u} - \nabla \cdot (\boldsymbol{\tau}_r - \boldsymbol{\tau}_{\text{sfs}}) \right) g dS. \quad (39)$$

The most significant difference between the one-phase VFIB method and the IB method introduced by Peskin, is that the present method intentionally accounts for the effect of smearing the IB. The thickness of the smearing is controlled using the filter thickness δ_f . Notice that, in the limit of vanishing filter thickness ($\delta_f \rightarrow 0$), we recover Peskin's equations as the unclosed terms in equation (39) vanish and the filter kernel g becomes a Dirac delta ($g \rightarrow \delta$). However, there are many advantages to choosing non-zero filter thickness. From a modeling perspective, this enables us to account for the effect of under-resolved flow motion in the vicinity of the IB using sub-filter scale terms $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_{\text{sfs}}$, if models can be supplied. This also allows us to purposely coarsen the resolution in applications such as LES, by taking δ_f larger than some cut-off length scale of the flow. From a computational perspective, the choice of $\delta_f \neq 0$ removes the need to discretize and regularize a Dirac delta distribution. This also has the advantage of decoupling the choice of filter width δ_f from the choice of grid spacing Δx , or any other discretization parameter. It also offers several advantages in the calculation of stresses on the IB and volume fraction field as we shall see in section 3.

2.3. Volume fraction computation

Although the volume fractions α_1 and α_2 do not appear explicitly in the one-phase formulation (equations (22) and (23)), the computation of the volume fraction serves three goals: (i) distinguish between interior and exterior points, (ii) compute the total volume occupied by the immersed solid, and most importantly (iii) extract the respective contributions due to external and internal fluids $\mathbf{F}_{I,1}$ and $\mathbf{F}_{I,2}$ from the total immersed boundary force density $\mathbf{F}_I$ (see §2.4).

Rather than using the formal definition equation (9) which requires computationally expensive procedures to build an indicator function [4,17], we compute the external fluid volume fraction α_1 by solving a Poisson equation. To derive this equation, first, replace $\boldsymbol{\Lambda}$ with the identity tensor $\mathbf{I}$ in equation (12) to obtain the gradient of the external fluid volume fraction,

$$\nabla \alpha_1(\mathbf{x}, t) = \iint_{\mathbf{y} \in S_I} \mathbf{n} g(\mathbf{x} - \mathbf{y}) dS. \quad (40)$$

The Poisson equation for the volume fraction is obtained by taking the divergence of the equation above:

$$\nabla^2 \alpha_1(\mathbf{x}, t) = \nabla \cdot \iint_{\mathbf{y} \in S_I} \mathbf{n} g(\mathbf{x} - \mathbf{y}) dS. \quad (41)$$

The above Poisson equation derived by volume-filtering is similar to the Poisson equation used to build phase-indicator functions in simulations of bubbly flows with the front-tracking method of Unverdi and Tryggvason [45]. Equation (41) can be parallelized and solved efficiently using elliptic solvers, including in simulations where immersed boundaries move and require frequent updates of α_1 . In the examples provided in section 5, we use an algebraic multigrid method to solve equation (41) with Dirichlet boundary conditions on the domain boundaries. For simplicity, we solve for the volume fraction throughout the entire domain, although it would suffice to solve (41) in a narrow band of thickness δ_f around the solid-fluid interface similar to the approach of Unverdi and Tryggvason [45]. Note that in the case of static immersed boundaries, equation (41) needs to be solved only once. For forcibly moved IBs, the solver (equations (38) and (39)) does not require the volume fraction, and as such, the computation of the volume-fraction is carried out as part of the post-processing workflow. For cases of Fluid-Structure Interaction, such as freely-moving particles, the volume fraction is needed at each step in order to compute the force on the immersed solids and update their dynamics (see example in §5.5). The volume fraction of the internal region is found using $\alpha_2 = 1 - \alpha_1$.

2.4. Computing the force due to the external fluid

As discussed in the introduction, the role of the internal flow must be clarified. In the one-phase formulation, the internal flow is not an artificial byproduct of the forcing technique, but instead has physical meaning. In the present approach, we

explicitly consider hollow solids filled with the same fluid as the one they have been immersed into. As a result, if the solid moves, the internal fluid develops a non-zero velocity, causing additional stresses on the interface. Even if the solid is static, the average nature of the formulation makes it that solving the mixture equations (24) and (25) may lead to non-zero mixture velocities inside the hollow solid, especially if the interface is very diffuse (i.e., δ_f/δ_c is not $\ll 1$). Thus, it is essential to be able to isolate the hydrodynamic stresses due to the external fluid from the total hydrodynamic stresses due to both internal and external fluids.

If stresses due to the internal flow can be neglected, the force on the immersed solid would be

$$\iint_{S_I} \mathbf{n} \cdot \boldsymbol{\tau}_1 dS = \iiint_{\mathbb{R}^3} \mathbf{F}_{I,1} dV \quad (42)$$

$$\simeq \iiint_{\mathbb{R}^3} \mathbf{F}_I dV = - \iint_{\mathbf{y} \in S_I} \ell(\mathbf{y}) \left(\rho_f \frac{D}{Dt} \mathbf{u} + \nabla p - \mu_f \nabla^2 \mathbf{u} - \nabla \cdot (\boldsymbol{\tau}_r - \boldsymbol{\tau}_{\text{sfs}}) \right) \Big|_{\mathbf{y}} dS \quad (43)$$

This approximation may be acceptable for flows with static boundaries and high momentum, since the internal flow is generally comparatively very small.

A more accurate estimation of the external forces on the immersed solid can be obtained by isolating and calculating the stresses due to the external fluid directly. Starting from the momentum conservation equations (23) for the external flow ($i = 1$) and following the same approach as in section 2.2, we obtain the force due to the external flow,

$$\begin{aligned} \iint_{S_I} \mathbf{n} \cdot \boldsymbol{\tau}_1 dS \\ = - \iint_{\mathbf{y} \in S_I} \ell(\mathbf{y}) \left(\rho_f \alpha_1 \frac{D}{Dt} \bar{\mathbf{u}}_1 - \nabla \cdot \left(-\alpha_1 \bar{p}_1 \mathbf{I} + \mu_f \alpha_1 \left(\nabla \bar{\mathbf{u}}_1 + \nabla \bar{\mathbf{u}}_1^T - \frac{2}{3} (\nabla \cdot \bar{\mathbf{u}}_1) \mathbf{I} \right) + \alpha_1 \mathbf{R}_{\mu,1} - \alpha_1 \boldsymbol{\tau}_{\text{sfs},1} \right) \right) \Big|_{\mathbf{y}} dS \end{aligned} \quad (44)$$

This expression can be rearranged in the following way

$$\iint_{S_I} \mathbf{n} \cdot \boldsymbol{\tau}_1 dS = \iint_{\mathbf{y} \in S_I} \mathbf{n} \cdot (\bar{\boldsymbol{\tau}}_1 - \boldsymbol{\tau}_{\text{sfs},1}) dS - \iint_{\mathbf{y} \in S_I} \alpha_1(\mathbf{y}) \ell(\mathbf{y}) \left(\rho_f \frac{D}{Dt} \bar{\mathbf{u}}_1 - \nabla \cdot (\bar{\boldsymbol{\tau}}_1 - \boldsymbol{\tau}_{\text{sfs},1}) \right) \Big|_{\mathbf{y}} dS \quad (45)$$

where $\bar{\boldsymbol{\tau}}_1 = -\bar{p}_1 \mathbf{I} + \mu_f (\nabla \bar{\mathbf{u}}_1 + \nabla \bar{\mathbf{u}}_1^T - 2/3 \nabla \cdot \bar{\mathbf{u}}_1) + \mathbf{R}_{\mu,1}$ is the filtered stress tensor.

Since we solve for the mixture quantities ($\mathbf{u}$ and p), it is more advantageous to express (45) in terms of these quantities. This can be done when the immersed boundary is well-resolved. Under such condition, we may neglect the tensors $\boldsymbol{\tau}_{\text{sfs}}$ and $\mathbf{R}_{\mu}$, and make the approximations $\bar{\mathbf{u}}_1(\mathbf{y}) \simeq \mathbf{u}(\mathbf{y})$ and $\bar{p}_1(\mathbf{y}) \simeq p(\mathbf{y})$ for points $\mathbf{y}$ on the interface S_I . Thus, equation (45) becomes

$$\iint_{S_I} \mathbf{n} \cdot \boldsymbol{\tau}_1 dS \simeq \iint_{S_I} \left(-p \mathbf{I} + \mu_f (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \right) \cdot \mathbf{n} dS - \iint_{\mathbf{y} \in S_I} \alpha_1(\mathbf{y}) \ell(\mathbf{y}) \left(\rho_f \frac{d}{dt} \mathbf{u}_I + \nabla p - \mu_f \nabla^2 \mathbf{u} \right) \Big|_{\mathbf{y}} dS, \quad (46)$$

where we have also used the no-slip boundary condition in the form $D\mathbf{u}/Dt = d\mathbf{u}_I/dt$.

Equation (46) shows that, to get the force exerted on the immersed boundary, it is not enough to integrate the (resolved) stresses on the immersed boundary, i.e., computing the first term on the right-hand side of (46) only. Doing so would lead to significant under-estimation of the force due to the external fluid. This is, perhaps, the reason why prior investigators sought alternative ways to compute the stresses on the immersed boundary, for example, by converting $\iint_{S_I} \mathbf{n} \cdot \boldsymbol{\tau}_1 dS$ into a volumetric integral and involving the internal flow. Such methods are not needed if both terms in (46) are computed. This is the method that we use in the rest of the manuscript to calculate hydrodynamic forces on immersed boundaries.

3. Numerical implementation

The VFIB method is implemented in a library called LEAP and interfaced with the flow solver NGA. Below, we describe only elements pertaining to the implementation of the VFIB method. Details about other aspects of the flow solver NGA can be found in [10].

3.1. Spatial discretization of the interface

We now focus on the discretization of the forcing term in equation (37). Suppose that the immersed interface has been meshed, such that $S_I = \bigcup_{m=1}^N S_m$, where S_m are the elements of the mesh. Fig. 2 shows an example of a mesh for a spherical immersed boundary. The IB forcing can be written as the sum of discrete contributions from each mesh element

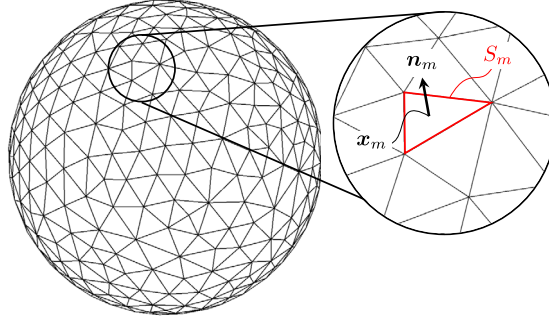


Fig. 2. Example of a mesh of a spherical immersed boundary. The “ m ”-th mesh triangle, S_m , has surface area A_m , centroid $\mathbf{x}_m$, and outward pointing normal $\mathbf{n}_m$.

$$\mathbf{F}_I(\mathbf{x}) = - \sum_{m=1}^N \iint_{\mathbf{y} \in S_m} \ell(\mathbf{y}) \left(\rho_f \frac{d}{dt} \mathbf{u}_I + \nabla p - \mu_f \nabla^2 \mathbf{u} - \nabla \cdot (\boldsymbol{\tau}_r - \boldsymbol{\tau}_{\text{sfs}}) \right) \Big|_{\mathbf{y}} g(\mathbf{x} - \mathbf{y}) dS + O(\delta_f^2). \quad (47)$$

Assuming that the typical mesh width is $O(\Delta x)$ and using the mid-point rule leads to

$$\mathbf{F}_I(\mathbf{x}) = - \sum_{m=1}^N \left\{ \ell(\mathbf{x}_m) \left(\rho_f \frac{d}{dt} \mathbf{u}_I + \nabla p - \mu_f \nabla^2 \mathbf{u} - \nabla \cdot (\boldsymbol{\tau}_r - \boldsymbol{\tau}_{\text{sfs}}) \right) \Big|_{\mathbf{x}_m} g(\mathbf{x} - \mathbf{x}_m) A_m \right\} + O(\Delta x^2, \delta_f^2), \quad (48)$$

where $\mathbf{x}_m$ is the location of the centroid of the mesh element S_m and A_m its surface area.

Expression (48) suggests a possible interpretation where the triangle centroids are viewed as Lagrangian forcing points, similar to the view adopted by Uhlmann [44]. With this interpretation, each Lagrangian forcing point in (48) can be associated with a volume $\Delta V_m = \ell(\mathbf{x}_m) A_m$. Thus, in the VFIB method, the Lagrangian point volume depends on the local curvature of the immersed boundary, choice of filter kernel g , and centroid spacing through the smearing length ℓ and the triangle surface area A_m . This is in contrast with the method of Uhlmann [44] where Lagrangian points are assigned a fixed volume $\Delta V_m = \Delta x^3$, regardless of the local topology of the immersed boundary. Further, the volume $\Delta V_m = \ell(\mathbf{x}_m) A_m$ can be computed directly from the surface density ($\Sigma = \ell^{-1}$) and the triangle surface area A_m without resorting to any optimization method as done by Zhou and Balachandar [49].

3.2. Temporal discretization

The time integration scheme is based on a semi-implicit iterative Crank-Nicolson scheme originally developed by Akselvoll and Moin [1] and Pierce and Moin [30] and recently adapted by Kasbaoui et al. [16] for simulations with immersed boundaries. The steps below describe the update from time t^n to t^{n+1} .

Step 1a: The first step consists in updating the immersed boundary to the $n+1$ time step. This is performed by updating the locations and velocities of the Lagrangian centroids to the new time step.

$$\mathbf{x}_m^n \rightarrow \mathbf{x}_m^{n+1}; \quad \mathbf{u}_{I,m}^n \rightarrow \mathbf{u}_{I,m}^{n+1} \quad (49)$$

In cases where the motion of the boundary is predetermined, the positions and velocities can be updated according to the laws of rigid body motion. A more general scheme can be used for applications in fluid-structure interaction.

Step 1b: Once the new location of the centroids is found, we compute the surface density Σ using:

$$\Sigma^{n+1}(\mathbf{x}) = \sum_m^N g(\mathbf{x} - \mathbf{x}_m^{n+1}) A_m \quad (50)$$

which represents equation (34) discretized with the mid-point rule. Section 3.3 provides details on the extrapolation procedure used to build the fields $g(\mathbf{x} - \mathbf{x}_m)$ in our Finite Volume solver.

Step 1c: Next, the smearing length at the centroids is obtained by taking the inverse of the surface density interpolated at the centroids,

$$\ell(\mathbf{x}_m^{n+1}) = (\Sigma^{n+1}(\mathbf{x}_m^{n+1}))^{-1}.$$

Interpolations are performed by taking convolutions with the filter kernel g . Details are provided in §3.3.

Step 1d: If desired, the new fluid volume fraction field α_1^{n+1} is computed by solving the discretized Poisson equation (41),

$$\nabla^2 \alpha_1^{n+1} = \nabla \cdot \left\{ \sum_m^N \mathbf{n}_m g(\mathbf{x} - \mathbf{x}_m) A_m \right\}. \quad (51)$$

Note that if the immersed boundary is static, steps 1a-d need only be performed at the simulation start.

Step 2: At this step the iterative loop is initiated. We assume that k sub-iterations have been performed and show the calculations for $(k+1)^{\text{th}}$ sub-iteration. As in [16], operator splitting is used to decouple momentum update, immersed boundary forcing, and pressure correction. At this step, the momentum update is performed without the immersed boundary force density:

$$\mathbf{u}_k^{n+1/2} = (\mathbf{u}_k^{n+1} + \mathbf{u}^n) / 2 \quad (52)$$

$$\tilde{\mathbf{u}}_{k+1}^{n+1} = \mathbf{u}^n + \Delta t \left(-\nabla \cdot (\mathbf{u}\mathbf{u})|_k^{n+1/2} - \frac{1}{\rho_f} p_k^{n+1} + \frac{\mu_f}{\rho_f} \nabla^2 \mathbf{u}_k^{n+1/2} + \frac{\partial \mathcal{M}}{\partial \mathbf{u}} \left(\frac{\tilde{\mathbf{u}}_{k+1}^{n+1} - \mathbf{u}_k^{n+1}}{2} \right) \right). \quad (53)$$

In the above, the operator $\mathcal{M}$ represents the sum of the convective and viscous operators,

$$\mathcal{M}(\mathbf{u}) = -\nabla \cdot (\mathbf{u}\mathbf{u}) + \frac{\mu_f}{\rho_f} \nabla^2 \mathbf{u}. \quad (54)$$

The Jacobian $\partial \mathcal{M} / \partial \mathbf{u}$ in equation (53) allows the treatment of the non-linearity with a Newton-Raphson method [30]. The momentum equation is solved using the approximate factorization technique of Choi and Moin [8] based on the Alternating Direction Implicit (ADI) method.

Step 3: Next, the immersed boundary term is applied. Using expression (48), the force density $\mathbf{F}_I$ is discretized as

$$\mathbf{F}_{I,k+1}^{n+1}(\mathbf{x}) = \sum_{m=1}^N \left\{ \ell(\mathbf{x}_m^{n+1}) \left(\frac{\mathbf{u}_{I,m}^{n+1} - \tilde{\mathbf{u}}_{k+1}^{n+1}(\mathbf{x}_m^{n+1})}{\Delta t} \right) g(\mathbf{x} - \mathbf{x}_m^{n+1}) A_m \right\}, \quad (55)$$

where we have used the fact that the velocity at $n+1$ at the centroid location must match the interface velocity at the new time step $\mathbf{u}_{I,m}^{n+1}$ to fulfill the no-slip boundary condition. The velocity field is then updated using,

$$\hat{\mathbf{u}}_{k+1}^{n+1} = \tilde{\mathbf{u}}_{k+1}^{n+1} + \Delta t \mathbf{F}_{I,k+1}^{n+1}. \quad (56)$$

Step 4: The pressure-Poisson equation is solved and a final velocity correction is applied to yield a divergence-free field,

$$\nabla^2 p_{k+1}^{n+1} = -\frac{\rho_f}{\Delta t} \nabla \cdot \hat{\mathbf{u}}_{k+1}^{n+1}, \quad (57)$$

$$\mathbf{u}_{k+1}^{n+1} = \hat{\mathbf{u}}_{k+1}^{n+1} - \frac{\Delta t}{\rho_f} \nabla (p_{k+1}^{n+1} - p_k^{n+1}). \quad (58)$$

Step 5: Repeat steps 2 to 4 until completion of the iterative Crank-Nicolson loop. Typically, two to three sub-iterations per time step are used [16].

3.3. Interpolations and extrapolations

In order to implement the VFIB method, there are two operations that require close examination: (a) *interpolations* of Eulerian quantities (e.g. fluid velocities) onto Lagrangian points (e.g. centroids of surface triangles), and (b) *extrapolations* of Lagrangian quantities (e.g. forcing at the centroids of surface triangles) onto the grid.

3.3.1. Interpolations

Interpolations are carried out by taking convolutions with the filter kernel g . For the sake of brevity, we illustrate how the calculation is performed in a one-dimension. Fig. 3 shows a sketch of the configuration.

For a Lagrangian point located at x_p , the interpolation of an Eulerian quantity $\Lambda(x)$ at the Lagrangian point is

$$\Lambda(x_p) = \int_{-\infty}^{+\infty} \Lambda(x') g(x' - x_p) dx'. \quad (59)$$

Taking into account the compactness of g , i.e., $g(x') = 0$ if $|x' - x_p| \geq \delta_f/2$, the interpolated quantity may be written as

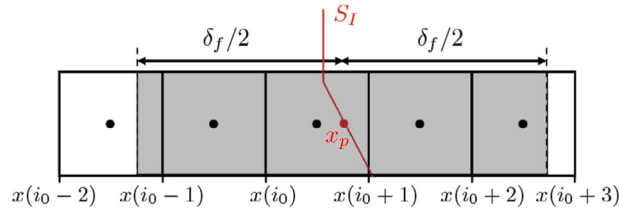


Fig. 3. Schematic showing the cells involved in the 1-D interpolations and extrapolations at a point x_p on the interface. In this example, the filter kernel has width $\delta_f = 4\Delta x$. The shaded area shows the cells that are under the reach of the filter.

$$\Lambda(x_p) = \int_{x_p - \delta_f/2}^{x_p + \delta_f/2} \Lambda(x') g(x' - x_p) dx'. \quad (60)$$

With a filter kernel $\delta_f/\Delta x = 4$, the integral above can be split into 5 contributions coming from cell i_0 containing the Lagrangian point and two neighboring cells on each side (see Fig. 3),

$$\Lambda(x_p) = \sum_{i=i_0-2}^{i_0+2} \int_{x(i)}^{x(i+1)} \Lambda(x') g(x' - x_p) dx'. \quad (61)$$

Finally, we use a mid-point rule to get a second order approximation of the interpolated value

$$\Lambda(x_p) = \frac{\sum_{i=i_0-2}^{i_0+2} \Lambda(x(i+1/2)) g(x(i+1/2) - x_p) \Delta x(i)}{\sum_{i=i_0-2}^{i_0+2} g(x(i+1/2) - x_p) \Delta x(i)} \quad (62)$$

where $\Delta x(i) = x(i+1) - x(i)$ is the size of cell i . Normalization in (62) is needed to ensure that the kernel is unitary in a discrete sense.

3.3.2. Extrapolations

Extrapolating a Lagrangian quantity λ_p defined at the Lagrangian point $\mathbf{x}_p$ consists in building the Eulerian field $\Lambda_p(\mathbf{x}) = \lambda_p g(\mathbf{x} - \mathbf{x}_p)$ on the fluid grid. For illustration, we will use the same one-dimensional configuration shown in Fig. 3.

We perform extrapolating operations in a Finite-Volume sense, i.e., we compute the cell averages

$$\Lambda_p(i) = \frac{1}{\Delta x(i)} \int_{x(i)}^{x(i+1)} \Lambda_p(x') dx' = \lambda_p \left(\frac{1}{\Delta x(i)} \int_{x(i)}^{x(i+1)} g(x' - x_p) dx' \right) \quad (63)$$

for $i = i_0 - 2, \dots, i_0 + 2$. For all other i , $\Lambda_p(i) = 0$ due to the compactness of the filter kernel. In order to reduce spurious force oscillations with moving immersed boundaries, we calculate the integrals above analytically. To do so, we use the following decomposition,

$$\int_{x(i)}^{x(i+1)} g(x' - x_p) dx' = \int_0^{x_r} g(x') dx' - \int_0^{x_l} g(x') dx', \quad (64)$$

where $x_r = \min(\max(x(i+1) - x_p, -\delta_f/2), \delta_f/2)$ and $x_l = \min(\max(x(i) - x_p, -\delta_f/2), \delta_f/2)$. With the filter kernels considered in this study, the integrals in (64) can be computed by hand with ease and directly implemented into the solver.

4. Characteristics of different filter kernels

The filter kernel plays a direct role in the immersed boundary force density, the surface density Σ , and the fluid/solid volume fraction. In the present work, we build the three-dimensional filter kernel g as the product of three 1-dimensional filter kernels

$$g(\mathbf{x} - \mathbf{y}) = g_1(|x_1 - y_1|) g_1(|x_2 - y_2|) g_1(|x_3 - y_3|). \quad (65)$$

We consider 4 different 1-dimensional filter kernels called triangle, parabolic, cosine, and triweight as described in Table 1. Fig. 4 shows a graph of these kernels. These filters are unitary, symmetric, and compact with width δ_f . The triweight filter has the highest peak and therefore the narrowest kernel with the majority of the weight distributed in the center, followed

Table 1
One-dimensional filter kernels considered in the present study.

Kernel	Expression
Triangle	$g_1(r) = \frac{2}{\delta_f} (1 - 2 r /\delta_f)$ if $ r \leq \delta_f/2$, 0 otherwise.
Parabolic	$g_1(r) = \frac{3/2}{\delta_f} (1 - (2r/\delta_f)^2)$ if $ r \leq \delta_f/2$, 0 otherwise.
Cosine	$g_1(r) = \frac{\pi/2}{\delta_f} \cos(\pi r/\delta_f)$ if $ r \leq \delta_f/2$, 0 otherwise.
Triweight	$g_1(r) = \frac{35/16}{\delta_f} (1 - (2r/\delta_f)^2)^3$ if $ r \leq \delta_f/2$, 0 otherwise.

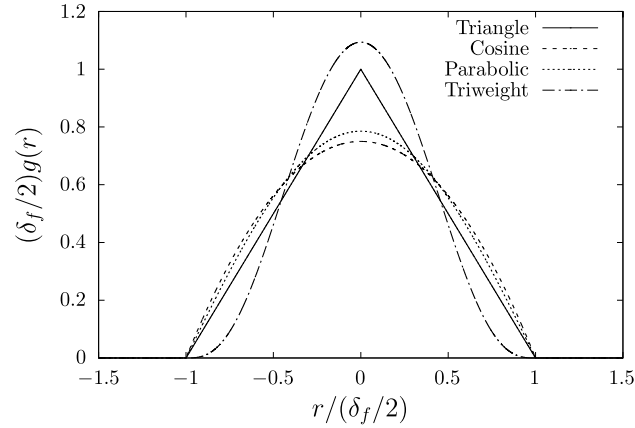


Fig. 4. Graph of the one-dimensional filter kernels in Table 1. The filter kernels are symmetric, integrate to unity, and have a compact support with width δ_f .

by the triangle kernel. Lastly, the cosine and parabolic kernel are similar in shape and have the smallest peak with the widest weight distribution.

Here, we emphasize that there is a fundamental difference between Peskin's regularized Dirac delta and the filter kernels considered here. The properties introduced by Peskin [29] originate from the fact that a true distribution cannot be represented on a discrete grid, thus, additional properties are needed to *regularize* the discrete Dirac delta. In contrast, the filter kernels discussed here, must satisfy only two conditions, being unitary and symmetric, to make the derivation of the volume-filtered equations possible. The third condition of compactness is added for computational efficiency, but other filters with infinite support (e.g. Gaussian filter) can be used in principle. With this distinction emphasized, we can also show that a unitary, symmetric, and compact filter kernel satisfies Peskin's properties in a discrete sense. Considering a 1D kernel, these three conditions yield

$$\text{compact:} \quad g_1(x) = 0 \quad \text{for} \quad |x| \geq \delta_f/2 \quad (66)$$

$$\text{unitary:} \quad \int_{\mathbb{R}} g_1 dx = 1 \rightarrow \sum_i g_1(x_i) \Delta x_i = 1 \quad (67)$$

$$\text{symmetric:} \quad g_1(-x) = g_1(x) \rightarrow \int_{\mathbb{R}} (x' - x) g_1(x' - x) dx = 0 \rightarrow \sum_i (x_i - x_{i_0}) g_1(x_i - x_{i_0}) \Delta x_i = 0 \quad (68)$$

where the integrals are discretized using the mid-point rule. Thus, the requirement that g is unitary, symmetric, and compact is sufficient to satisfy Peskin's conditions [29].

To illustrate the impact of these kernels, we consider a circular immersed boundary of diameter D in Fig. 5. The Cartesian grid is uniform with grid spacing $\Delta x = D/24$. In order to retain small stencils for interpolations and extrapolations, the ratio of filter width to mesh spacing is set to a fixed value $\delta_f/\Delta x = 4$. This gives a moderate resolution of the cylinder as shown by the ratio $\delta_f/D = 1/6$. The top panels in Fig. 5 show the surface density Σ obtained using the four filter kernels in Table 1. The surface density field reflects clearly the relative spread of the filter kernel, whereby the narrowest kernel, the triweight kernel here, leads to the largest surface density peaks. Conversely, the parabolic kernel, which is the most spread out, leads to comparatively lower surface density. The bottom panels in Fig. 5 shows the fluid volume fraction field obtained by solving the Poisson equation (41). As expected, inside the fluid volume fraction is equal to 0 inside the cylinder and equal to 1 outside. Near the boundary, the volume fraction transitions smoothly. This transition is slightly sharper for the narrower filter kernels, although these differences are not as significant as those seen for surface density.

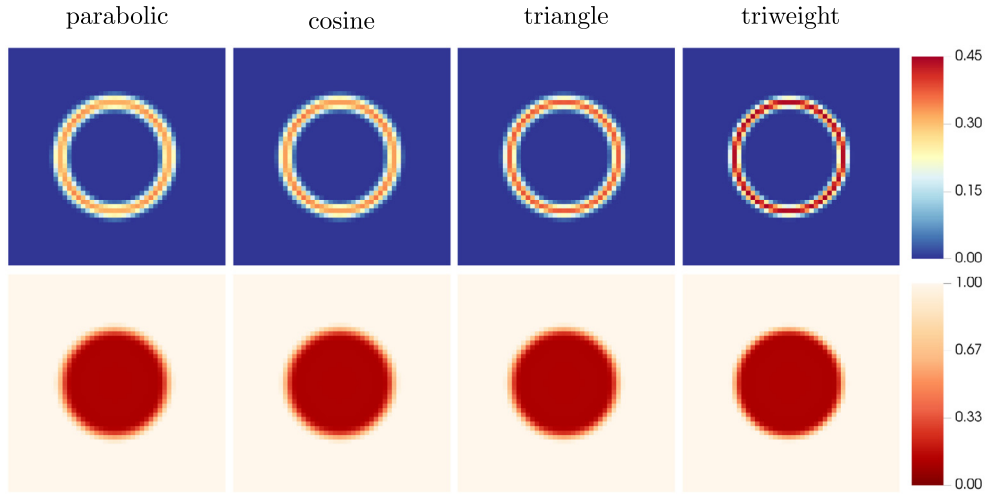


Fig. 5. The effect of varying filter kernels on the (top) normalized surface density $\Sigma/(\pi DLz/\delta_f^3)$ and (bottom) fluid volume fraction α_1 for a circular immersed boundary of diameter D with filter width $\delta_f/D = 1/6$. The grid spacing is such that 4 grid points lie across the filter width ($\delta_f/\Delta x = 4$). (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

5. Test cases using the VFIB method

In this section, we apply the VFIB method in five benchmark tests with static and moving immersed boundaries. The rationale for the choice of these cases is as follows:

- **Fixed cylinder in a channel flow:** This two-dimensional case demonstrates the VFIB in a simple configuration. We use this case to investigate the effect of varying (i) filter kernels, (ii) varying filter sizes D/δ_f , and (iii) resolutions ($D/\Delta x$ and $\delta_f/\Delta x$).
- **Fixed cylinder in free stream:** We consider this second two-dimensional test case to enable comparison with other immersed boundary methods since this is among the most popular benchmark tests.
- **Laterally oscillating cylinder in crossflow:** This case demonstrates the ability of the VFIB method to handle forcibly moving IBs in 2D.
- **Flow past a sphere:** This case shows the ability of the method to handle 3D static IBs, and reproduce intricate wake patterns.
- **Freely falling sphere under gravity:** This case shows the ability of the method to handle freely moving 3D IBs. It shows the potential of the method in applications related to Particle-Resolved DNS and Fluid-Structure Interaction (FSI).

5.1. Fixed cylinder in a channel flow

In this first test, we consider the two-dimensional case of a static cylinder of diameter D in a channel at Reynolds number $Re_D = 100$. The channel length and height are $L_x = 22D$ and $L_y = 4.1D$, respectively. The cylinder is placed asymmetrically at $x = y = 0.3D$. A parabolic inflow with average velocity U is prescribed at the inlet $x = 0$. The average velocity U and fluid kinematic viscosity ν are chosen such that $Re_D = UD/\nu = 100$.

In the VFIB method, we make a distinction between the IB resolution, characterized by the ratio δ_f/D , and the grid resolution, characterized by the ratio $D/\Delta x$, although the two are connected. We consider three levels of resolution of the immersed boundary: (i) coarse resolution with $\delta_f/D = 1/6$, (ii) medium resolution with $\delta_f/D = 1/12$, and (iii) high resolution with $\delta_f/D = 1/24$. Unless noted otherwise, the grid spacing is chosen such that $\delta_f/\Delta x = 4$. This choice has the benefits of providing sufficient resolution of the filter kernel while retaining compact stencils for interpolations and extrapolations that use only 5 grid points in each direction. With fixed $\delta_f/\Delta x = 4$, the coarse, medium, and fine IB resolutions yield increasingly fine grids characterized by the ratios $D/\Delta x = 24, 48$, and 96.

Fig. 6a shows isocontours of normalized velocity magnitude $\|\mathbf{u}\|/U$ at the medium IB resolution $\delta_f/D = 1/12$ obtained using the triangle filter kernel. The wake behind the cylinder shows a pair of attached eddies that form behind the cylinder similar to the observations made by Tritton [41] and Zdravkovich [48]. Fig. 6a also shows the existence of an internal flow within the cylinder. This internal flow is anticipated in the one-phase formulation because $\mathbf{u} = \alpha_1 \bar{\mathbf{u}}_1 + \alpha_2 \bar{\mathbf{u}}_2$ represents the sum of both internal and external flows. The former flow can be hidden by multiplying the total fluid velocity with the external fluid volume fraction α_1 , as shown in Fig. 6b, since $\alpha_1 \mathbf{u} \simeq \alpha_1 \bar{\mathbf{u}}_1$.

To assess quantitatively the performance of the VFIB method, we compare with the benchmark study of Schäfer et al. [34] where the results from several simulations using body-fitted meshes are compiled. For the present case at $Re_D = 100$, Schäfer et al. [34] give a Strouhal number $St = 0.3 \pm 0.005$, maximum drag coefficient $C_{D,\max} = 3.23 \pm 0.01$, and maximum

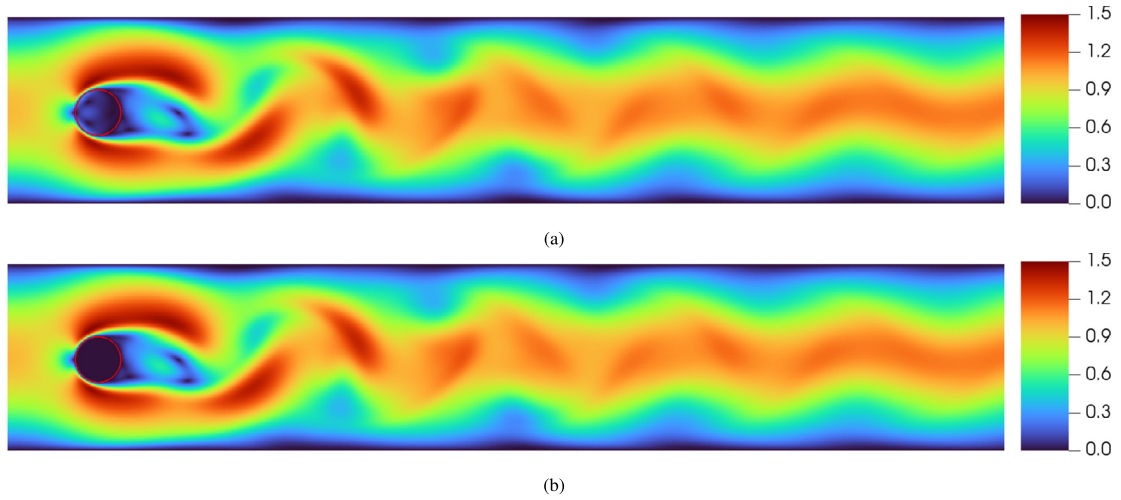


Fig. 6. Isocontours of (a) $\|\mathbf{u}\|/U$ and (b) $\|\alpha_1 \bar{\mathbf{u}}_1\|/U$ at the resolution $\delta_f/D = 1/12$ and using the triangle filter kernel. The location of the immersed boundary is identified by the isolevel $\alpha_1 = \alpha_2 = 0.5$. Multiplying by the volume fraction of the external fluid α_1 hides the internal flow.

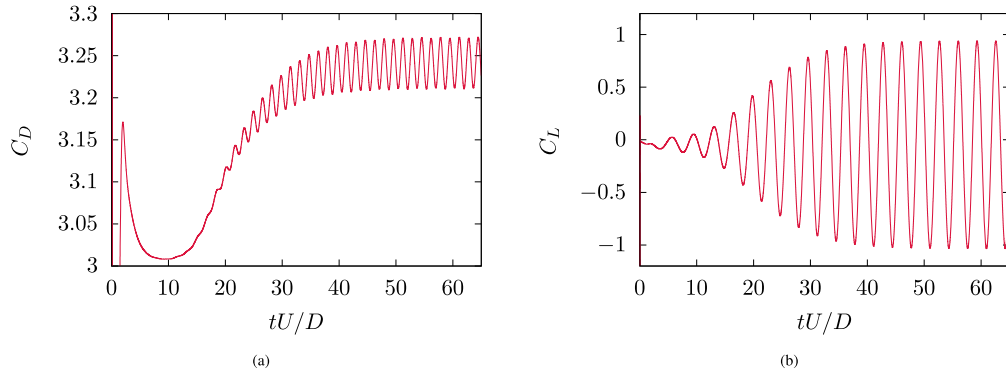


Fig. 7. Time evolution of the (a) drag coefficient and (b) lift coefficient for the case of a cylinder asymmetrically placed in a channel at $Re_D = 100$. The data is obtained using the triangle filter kernel with width $\delta_f/D = 1/24$.

lift coefficient $C_{L,\max} = 1.0 \pm 0.01$. In the present work, drag and lift forces on the immersed solid are computed using the contribution of the external fluid only as explained in §2.4. Fig. 7 shows the evolution of the drag and lift coefficients at the resolution $\delta_f/D = 1/24$ using the triangle kernel. These quantities reach a stationary state after $t \sim 40D/U$. We compute statistics using data from this point until $t \sim 65D/U$ (about 15 periods). For the case shown in Fig. 7, we find $St = 0.300$, $C_{D,\max} = 3.27$, and $C_{L,\max} = 0.94$. These values are in excellent agreement with the results of body-fitted simulations reported by Schäfer et al. [34].

Fig. 8 shows that increasing the IB resolution causes $C_{D,\max}$, $C_{L,\max}$, and the Strouhal number St to converge to the benchmark values, regardless of the choice of filter kernel. Note that with improving IB resolution, i.e., decreasing ratio δ_f/D , more grid points are required to maintain the ratio $\delta_f/\Delta x = 4$. From Fig. 8, we observe that the predicted drag, lift, and Strouhal converge to the benchmark values as δ_f/D is reduced for all filter kernels considered. The choice of filter kernel has little effect on the predicted maximum drag and lift coefficients, and Strouhal number provided that the immersed boundary is sufficiently well resolved.

For a fixed filter width δ_f , the predicted drag and lift coefficients also converge with decreasing Δx . In Fig. 9, the filter width is maintained at $\delta_f/D = 1/12$, while the grid spacing Δx progressively reduced such that the ratio $\delta_f/\Delta x$ varies from 2 to 8. At the lower end $\delta_f/\Delta x = 2$, the interpolation/extrapolation require stencils with only 3 grid points in each direction, thus, making this choice the most computationally efficient one. At $\delta_f/\Delta x = 8$, the stencils require 9 grid points in each direction. As shown in Fig. 9, $C_{D,\max}$, $C_{L,\max}$ converge with increasing ratio $\delta_f/\Delta x$. The choice of $\delta_f/\Delta x = 4$ is a happy medium between small stencils and well converged results.

5.2. Fixed cylinder in free stream

We now consider the case of a fixed cylinder of diameter D placed in free stream. Unless otherwise noted, the simulations are carried out in a computational domain of size $L_x = L_y = 26D$, similar to the configuration considered by

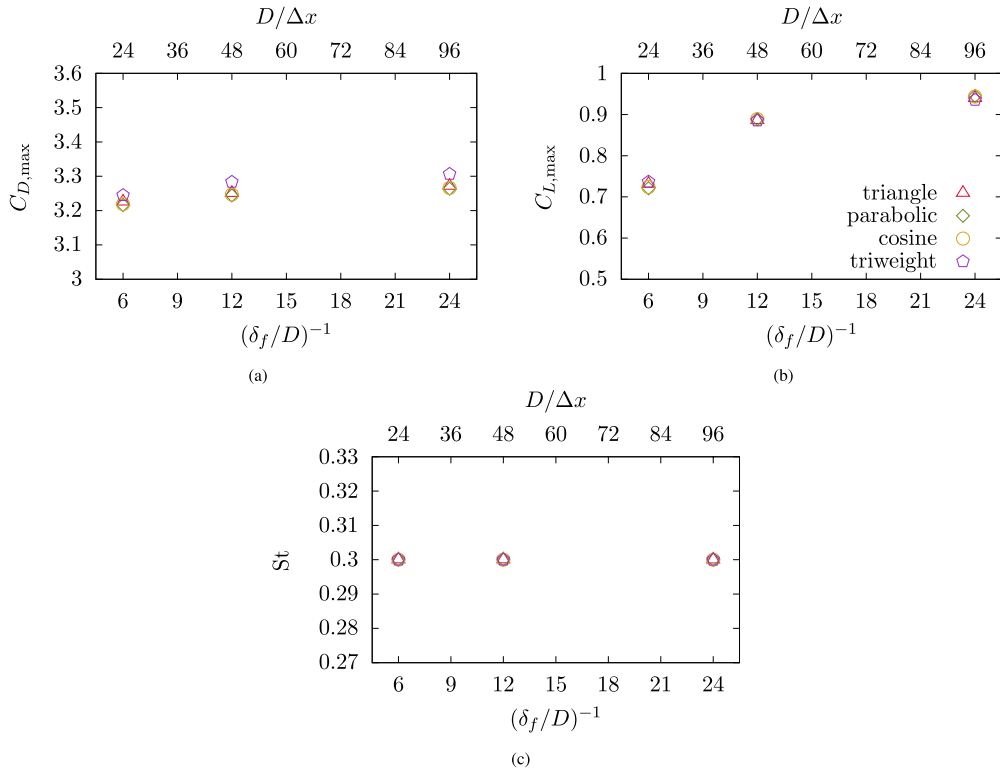


Fig. 8. Effect of varying resolution and filter kernels on the (a) maximum drag coefficient, (b) maximum lift coefficient, and (c) Strouhal number for the case of a cylinder asymmetrically placed in a channel at $Re_D = 100$.

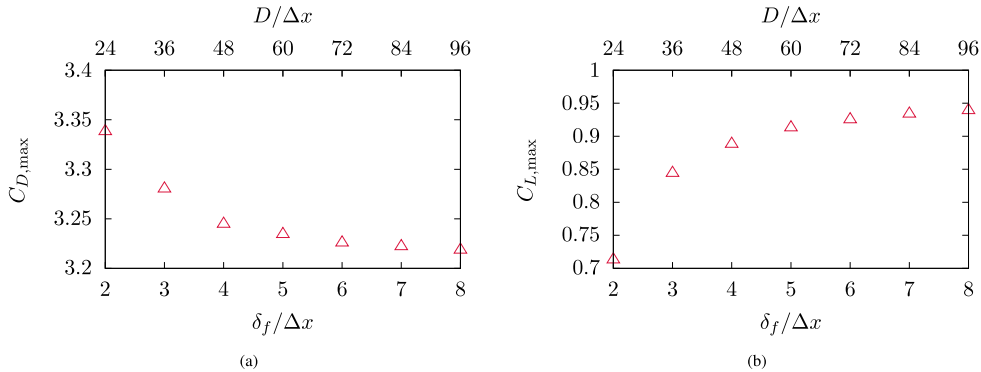


Fig. 9. Effect of increasing the grid resolution at fixed filter width $\delta_f = D/12$ on the (a) maximum drag coefficient, (b) maximum lift coefficient on a cylinder asymmetrically placed in a channel at $Re_D = 100$. For these simulations, the triangle filter kernel is used. Results are well converged for $\delta_f/\Delta x \geq 4$.

Uhlmann [44]. A uniform inflow with velocity U is prescribed at the inlet $x = 0$. The cylinder is placed at $x = 6D$ and centered in the y direction. The parameters are chosen such that the Reynolds number $Re_D = UD/\nu$ is 100. We also consider an additional simulation in an enlarged domain of size $L_x = L_y = 40D$ to compare with the sharp-interface IB method of Mittal et al. [24].

Once the flow over the immersed cylinder is established, a vortex street is observed in the wake of the cylinder. Fig. 10a shows that vortices are shed from the top and bottom of the cylinder at alternating intervals. This leads to oscillating drag and lift forces as shown in Figs. 10b and 10c. The periodic vortex shedding is characterized by the Strouhal number $St = f_0 D/U$ where f_0 is the shedding frequency extracted from the lift force.

Table 2 contains a summary of the Strouhal number St , mean drag coefficient $\bar{C}_D$, fluctuating drag coefficient C'_D , and fluctuating lift coefficient is C'_L from the present simulations with the triangle filter kernel. Data obtained with the body-fitted mesh simulations of Liu et al. [21], and IB methods of Uhlmann [44] and Mittal et al. [24] are also included for comparison. For the case shown in Fig. 10, with resolution $\delta_f = D/24$, we find $St = 0.165$, $\bar{C}_D = 1.434$, $C'_D = 0.010$, and $C'_L = 0.346$. For comparison, Liu et al. [21] give $St = 0.165$, $\bar{C}_D = 1.35$, $C'_D = 0.012$, and $C'_L = 0.339$ from simulations with

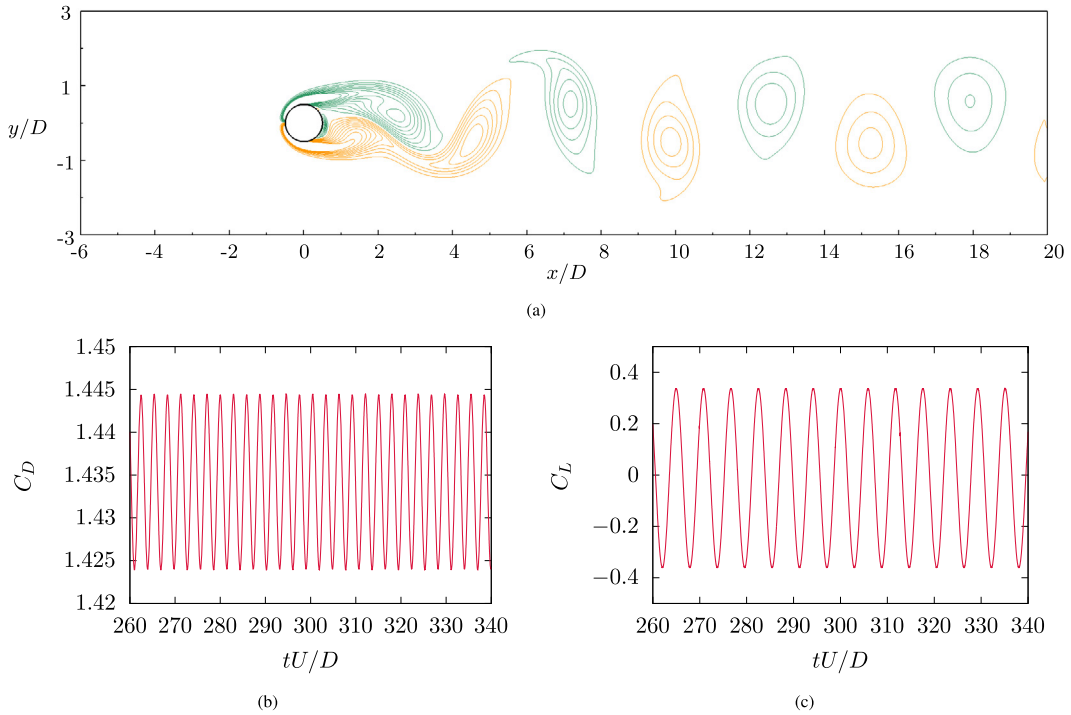


Fig. 10. Flow past an immersed cylinder at $Re_D = 100$ obtained using the VFIB method with the triangle filter kernel and width $\delta_f = D/24$. Vortex shedding can be seen from (a) alternating isolevels of positive (orange) and negative (green) vorticity. This results in fluctuating (b) drag and (c) lift coefficients.

Table 2

Characteristics of drag and lift coefficients for a cylinder in a freestream at $Re_D = 100$. The results from the VFIB method are compared to body-fitted simulations of Liu et al. [21] and the immersed boundary approach of Uhlmann [44] and Mittal et al. [24].

	δ_f/D	$D/\Delta x$	$\bar{C}_D$	C'_D	C'_L	St
Liu et al. [21]	–	–	1.350	0.012	0.339	0.165
Present	1/3	10	1.364	0.003	0.214	0.165
	1/5	20	1.424	0.008	0.320	0.165
	1/10	40	1.434	0.010	0.348	0.165
Present (enlarged)	1/16	64	1.355	0.010	0.331	0.165
Uhlmann [44]	–	38.4	1.501	0.011	0.349	0.172
Mittal et al. [24]	–	66	1.350	–	–	0.165

body-fitted mesh, whereas Uhlmann [44] gives $St = 0.172$, $\bar{C}_D = 1.501$, $C'_D = 0.011$, and $C'_L = 0.349$ using his immersed boundary method. Thus, the agreement with the benchmark results of Liu et al. [21] is improved using the VFIB method, particularly for the Strouhal number and mean drag coefficient. The latter is within 6.2% of the value reported by Liu et al. [21], compared to an over-prediction by 11.1% given by Uhlmann [44].

The drag over-prediction reduces further to less than 0.37% when the simulation is carried in the enlarged domain and at the higher resolution $\delta_f/D = 1/16$. This resolution, which corresponds to $D/\Delta x = 64$, and enlarged domain are chosen to approximately match the domain and resolution used by Mittal et al. [24], that is $D/\Delta x = 66$. As shown in Table 2, our results match very closely those of Mittal et al. [24] and the reference results of Liu et al. [21].

Fig. 11 shows the effect of varying the filter kernel and width. Similar to what we have shown in the previous numerical example, these results show that the predicted St , $\bar{C}_D$, C'_D , and C'_L converge with improving resolution of the immersed cylinder and that the choice of the filter kernel has little impact if the immersed boundary is well resolved.

5.3. Laterally oscillating cylinder in a uniform crossflow

In this test, we consider a laterally oscillating immersed cylinder in a uniform crossflow at $Re_D = 185$. The configuration is identical to the one described in the previous test (§5.2), with the difference that, now, the cylinder has forced oscillations around its position. The displacement of the cylinder center is $\Delta y_c = 0.2D \sin(2\pi f_e t)$, where $f_e = 0.8f_0$ is the forced

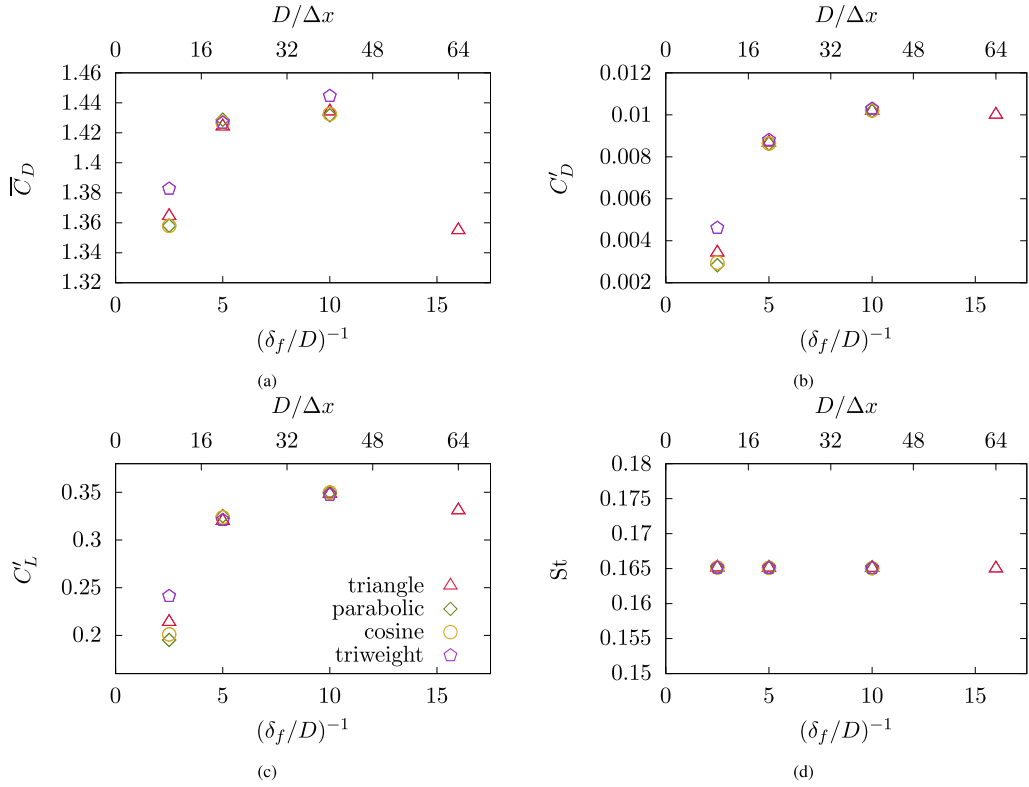


Fig. 11. Effect of varying resolution and filter kernel on the (a) mean drag, (b) drag fluctuation, (c) lift fluctuation, and (d) Strouhal number for the case of an immersed cylinder at $Re_D = 100$.

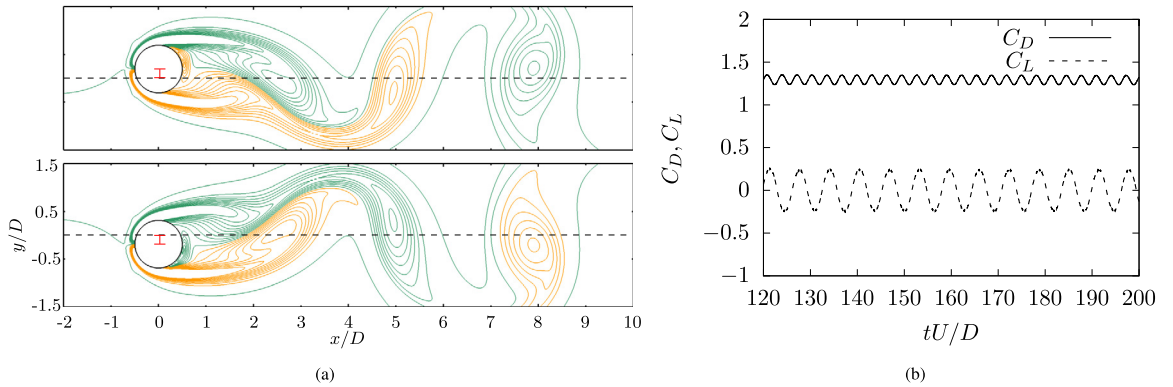


Fig. 12. Flow past an oscillating immersed cylinder with a crossflow at $Re_D = 185$ obtained using the VFIB method with the triangle filter kernel, width $\delta_f/D = 1/24$, and $CFL_{\max} = 0.5$. (a) Isocontours of positive (orange) and negative (green) vorticity when the cylinder is at its highest (top) and lowest (bottom) positions. This results in fluctuations of (b) drag and lift coefficients.

oscillation frequency, and f_0 is the natural shedding frequency at $Re_D = 185$. We consider two spatial resolutions at $\delta_f/D = 1/6$ (coarse), and $1/12$ (medium), and vary the time step Δt to yield a maximum Courant-Friedrich-Levy number $CFL_{\max}$ between 0.5 and 0.1. Since we have determined that the choice of filter kernel plays little role for well-resolved immersed boundaries, we perform tests with the triangle kernel only.

Once the flow reaches a stationary state, a vortex street is observed in the wake of the cylinder. Fig. 12a shows the vortex contours when the cylinder is at its highest and lowest displacements. Compared to the case of a static cylinder, the vortex contours from the upper end of the cylinder are elongated, and the contours form the base of the cylinder are tightened, when the cylinder is in the highest displacement. This tightening of the base vorticity is due to the upward motion of the oscillating cylinder generating vorticity at the base. The opposite behavior is observed in the configuration of lowest displacement. Similar observations were also reported by Lu and Dalton [22] who used simulations with body-conforming

Table 3

Characteristics of drag and lift on a transversely oscillating cylinder at $Re_D = 185$. Results from the present VFIB method are compared to results of Lu and Dalton [22] obtained using body-confirming grid, and the results with the immersed boundary method of Uhlmann [44] and smoothing technique of Yang et al. [47]. For the latter, we report only their smoothest case obtained using their smoothed 4-point piecewise Dirac delta.

	δ_f/D	$D/\Delta x$	$CFL_{\max}$	$\bar{C}_D$	$C_{D,rms}$	$C_{L,rms}$
Lu and Dalton [22]	–	–	–	1.25	0.040	0.18
Present	1/6	24	0.5	1.28	0.031	0.135
	1/6	24	0.25	1.24	0.039	0.155
	1/6	24	0.1	1.24	0.039	0.177
	1/12	48	0.5	1.29	0.040	0.18
	1/12	48	0.25	1.25	0.040	0.18
Uhlmann [44]	–	38.4	0.6	1.380	0.045	0.176
Yang et al. [47]	–	50	–	1.29	0.043	0.07

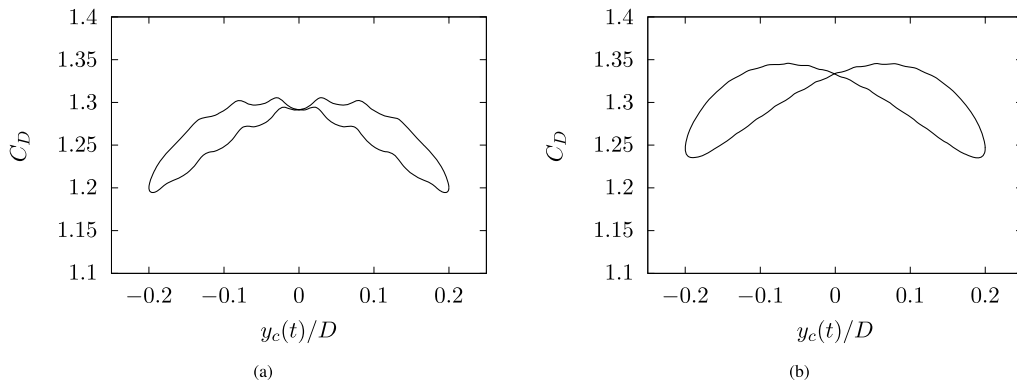


Fig. 13. Time-periodic variation of C_D with respect to the position of the oscillating cylinder for $Re_D = 185$ at (a) $CFL_{\max} = 0.5$, $\delta_f/D = 1/6$ ($D/\Delta x = 24$) and (b) $CFL_{\max} = 0.5$, $\delta_f/D = 1/12$ ($D/\Delta x = 48$).

meshes. As a result of this vortex shedding, oscillatory fluctuations in the coefficient of drag and lift are witnessed as shown in Fig. 12b.

Table 3 contains a summary of the mean drag coefficient $\bar{C}_D$, drag fluctuations $C_{D,rms}$, and lift fluctuations $C_{L,rms}$. The values reported by Lu and Dalton [22] and Uhlmann [44] are also reported for comparison.

At the coarse resolution $\delta_f/D = 1/6$, there is a strong sensitivity to temporal errors, but, these reduce significantly with decreasing $CFL_{\max}$. Taking the values reported by Lu and Dalton [22] as reference, the VFIB method at $CFL_{\max} = 0.5$ yields $\bar{C}_D$, $C_{D,rms}$ and $C_{L,rms}$ that are within 2.5%, 22.8%, and 25% respectively, of the reference values. Reducing $CFL_{\max}$ to 0.25 or 0.1 causes deviations in predicted $\bar{C}_D$ and $C_{D,rms}$ to reduce to 1% and 3.5%, respectively. In addition, the deviations in $C_{L,rms}$ reduce to 14% and 1.8% at $CFL_{\max} = 0.25$ and 0.1. Note that even at $CFL_{\max} = 0.5$, and the coarse resolution $\delta_f/D = 1/6$ ($D/\Delta x = 24$), the values predicted using the VFIB method agree better with the reference values in [22] than those given by Uhlmann [44] with $D/\Delta x = 38.4$ and $CFL_{\max} = 0.6$ (see Table 3).

With improving resolution of the immersed boundary to $\delta_f/D = 1/12$, the predicted $C_{D,rms}$ and $C_{L,rms}$ converge to those given by Lu and Dalton [22] even at $CFL_{\max} = 0.5$. The mean drag coefficient $\bar{C}_D$ is off by 3.2% at $CFL_{\max} = 0.5$ and converges to the exact reference value at $CFL_{\max} = 0.25$.

This greatly enhanced performance compared to the method of Uhlmann [44] is the result of three improvements: (i) a more accurate way of computing the hydrodynamic force using equation (46), (ii) the fact that, in Uhlmann's method, the surface markers have a Lagrangian volume $\Delta V_m = \Delta x^3$ which, as we discussed at the end of section 3.1, is not accurate, and (iii) significantly reduced spurious force oscillations, as we discuss next.

Here, we emphasize that our treatment of extrapolations using analytical calculations of the integrals shown in §3.3 reduces spurious force fluctuations considerably without the need for any smoothing technique like the one developed by Yang et al. [47]. Fig. 13 shows a graph of C_D during 1 period of oscillation at $CFL_{\max} = 0.5$ at the resolutions $\delta_f/D = 1/6$ and $1/12$. Similar to what has been reported by Yang et al. [47] and Uhlmann [44], spurious force fluctuations can be seen in the graph with the coarse resolution $\delta_f/D = 1/6$ shown in Fig. 13a. These oscillations have length scale comparable to the mesh spacing Δx , and depend largely on how interpolations and extrapolations are performed numerically. Increasing the spatial resolution reduces the amplitude of the spurious oscillations significantly. Without using any smoothing technique, the curve in Fig. 13b ($\delta_f/D = 1/12$) is as smooth as the one shown in [47] obtained with similar numerical parameters.

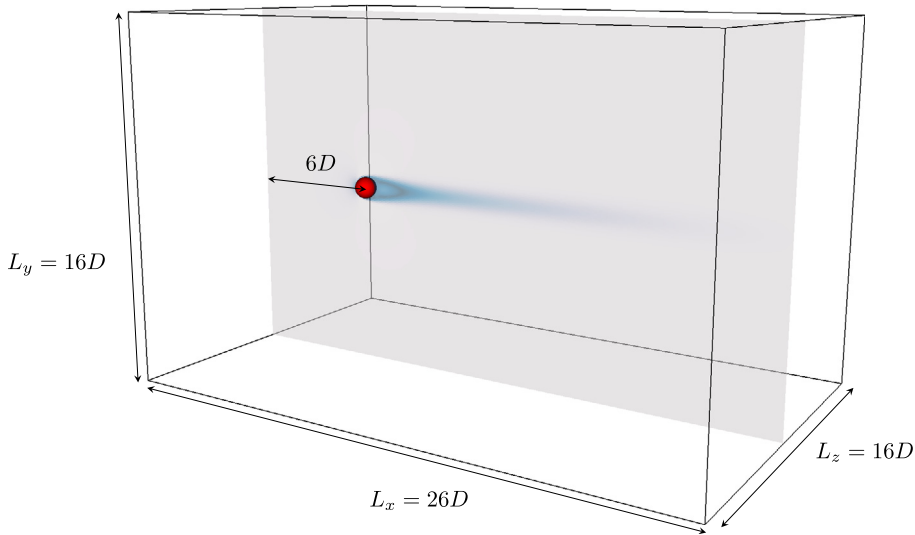


Fig. 14. Computational domain used for simulations of flow past a sphere.

This is because, as the immersed solid moves, cells that are only partially covered by the filter kernel are still forced when the extrapolation is performed as shown in §3.3, even if this coverage is minute. In contrast, in methods based on numerical approximations of the integrals using the mid-point rule, a cell would not be forced until at least half of it is covered by the kernel (or, similarly, the support of the discretized Dirac delta), which leads to jagged and discontinuous forcing as the immersed solid moves.

5.4. Flow past a sphere

We now consider a three dimensional case involving the flow past a sphere. This flow has been studied in detail by numerous investigators, thus, providing a wealth of information to validate the present immersed boundary strategy. Notably, Johnson and Patel [15] used a numerical approach based on a body-fitted grid to resolve the steady and unsteady dynamics associated with the transition from attached wake to vortex shedding in the wake of a sphere up to Reynolds number of 300. Johnson and Patel [15] showed that the wake is steady and axisymmetric up to $Re_D \simeq 200$. For Re_D between 210 and 270, the wake remains steady although the axisymmetry is lost. Values of Re_D greater than 270 lead to periodic vortex shedding in the form of a sequence of hairpin vortices.

In order to compare with the results of Johnson and Patel [15], we perform simulations of the flow over a sphere at Reynolds numbers from $Re_D = 25$ to 300. The simulation domain extends by $L_x = 26D$ in the flow direction, and $L_z = L_y = 16D$ in the two normal directions. Fig. 14 shows a schematic of the configuration. Since we have established that the choice of filter kernel has little effect in well-resolved simulations, we consider simulations with the triangle filter kernel only.

Fig. 15 shows the variation of the drag coefficient with Reynolds number. For comparison, data obtained using the VFIB method is plotted alongside results of the body-fitted grid simulations of Johnson and Patel [15], and experiments of [33]. Further, the Schiller-Naumann correlation $C_D^{SN} = \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687})$ is also reported on the same plot [35]. At low Reynolds numbers, the low resolution simulations with $\delta_f/D = 1/3$ give a drag coefficient in excess by up to 10% of the Schiller-Naumann drag coefficient, but still within good agreement with other numerical and experimental data. Increasing the resolution of the immersed boundary to $\delta_f/D = 1/6$ and $\delta_f/D = 1/9$ causes the predicted drag values to converge. At the highest resolution, there is excellent agreement with prior data and the Schiller-Naumann correlation.

In Fig. 16, we examine the characteristics of the wake up to $Re_D = 200$. Fig. 16a shows the streamlines in the x - y plane going through the sphere's center for the four cases at $Re_D = 50, 100, 150$, and 200, and at the resolution $\delta_f/D = 1/9$. At these Reynolds numbers, the flow is steady and axisymmetric. The near-wake flow exhibits a recirculation bubble similar to previously reported observations [40,31,15]. As shown in Fig. 16a, flow separation happens at an angle Θ_s that decreases with increasing Reynolds number, while the bubble length l_s increases with increasing Reynolds number. In Figs. 16b and 16c, we compare the values of bubble length l_s and separation angle Θ_s obtained with the VFIB method with those obtained by Johnson and Patel [15]. Fig. 16b shows that the resolution $\delta_f/D = 1/6$ is sufficient to yield excellent agreement with the data of Johnson and Patel [15]. However, capturing the separation angle Θ_s accurately requires higher resolution. For the cases with $\delta_f/D = 1/6$, we find a separation angle that is larger than the previously reported values by about 8 degrees at $Re_D = 25$, and 18 degrees at $Re_D = 200$. These deviations reduce to 1 degree at $Re_D = 25$, and 8 degrees at $Re_D = 200$, when the resolution is increased to $\delta_f/D = 1/9$. The trend in Fig. 16c shows that the angles obtained using the VFIB method converge towards the reference values, but higher resolution is required to achieve agreement within a few degrees.

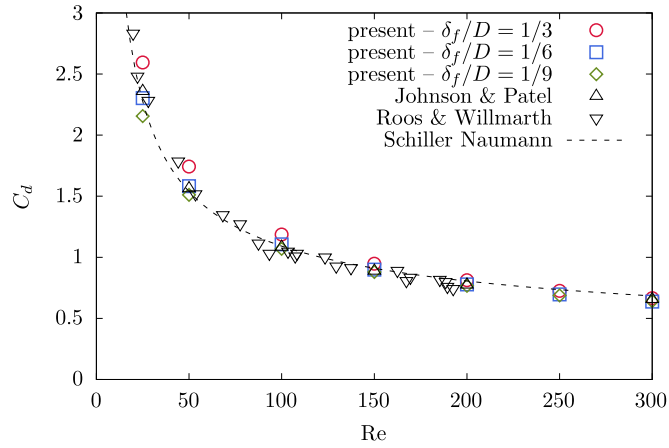


Fig. 15. Variation of the drag coefficient with Reynolds number for the flow past a sphere.

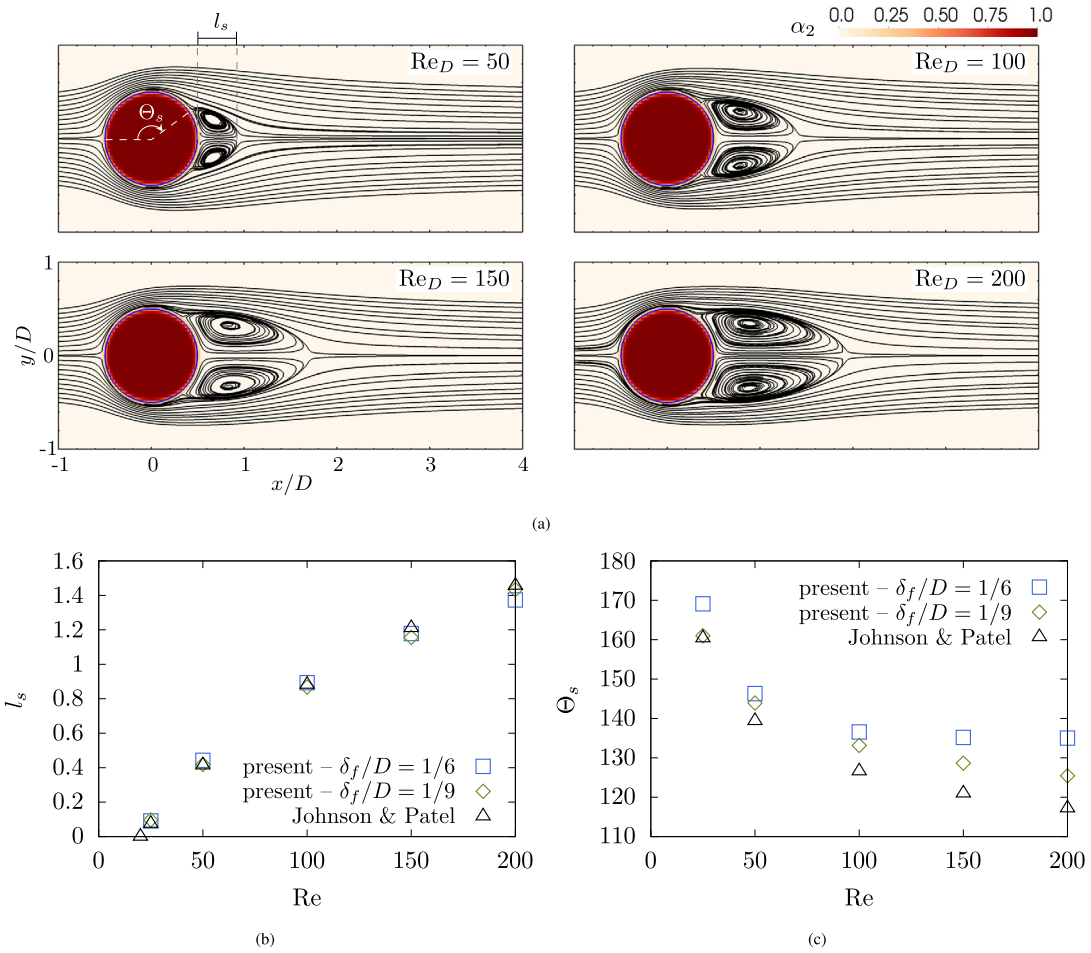


Fig. 16. Characteristics of the wake behind an immersed sphere in the axisymmetric regime: (a) streamlines at $Re_D = 50, 100, 150$, and 200 , (b) length of the separation bubble, and (c) separation angle.

When the Reynolds number is increased to $Re_D = 250$, the VFIB method correctly captures the transition to an asymmetric near-wake. Fig. 17 shows flow streamlines at $Re_D = 250$ and resolution $\delta_f/D = 1/9$ in the x - y and x - z planes. While the flow remains steady, an azimuthal mode appears in the near wake. This mode is symmetric about the x - z plane. These

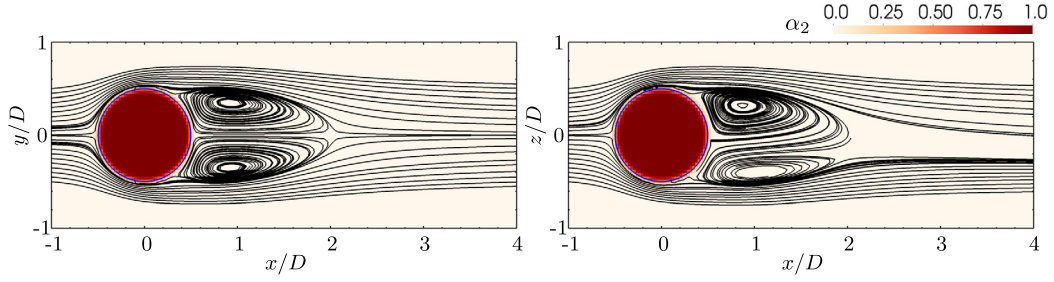


Fig. 17. Streamlines showing the asymmetric near-wake flow past a sphere at $Re_D = 250$.

observations are in agreement with the results of Johnson and Patel [15] who also observe loss of axisymmetry due to the emergence of an azimuthal mode with planar symmetry similar to what is shown in Fig. 17.

5.5. Freely falling sphere under gravity

In this final test, we consider the case of a freely falling sphere under gravity. The particle position $\mathbf{x}_p$, velocity $\mathbf{u}_p$, and angular velocity $\boldsymbol{\omega}_p$ are updated at each time step by solving the following equations of motion,

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p \quad (69)$$

$$m_p \frac{d\mathbf{u}_p}{dt} = \iint_{S_I} \mathbf{n} \cdot \boldsymbol{\tau}_1 dS + (\rho_p - \rho_f) \frac{\pi D^3}{6} \mathbf{g}_v \quad (70)$$

$$I_p \frac{d\boldsymbol{\omega}_p}{dt} = \iint_{S_I} (\mathbf{y} - \mathbf{y}_p) \times \mathbf{n} \cdot \boldsymbol{\tau}_1 dS \quad (71)$$

where ρ_p , D , $m_p = \rho_p(\pi/6)D^3$, and $I_p = m_p D^2/10$ are the particle density, diameter, mass, and moment of inertia. Here, $\mathbf{g}_v$ denotes the gravitational acceleration. The first term on the right-hand side of (70) represents the hydrodynamic stresses exerted by the external fluid and computed using the expression (46). This requires an update of the volume fraction field α_f at each step, which is carried out as described in section 2.3.

We use the experiments from Mordant and Pinton [26] as benchmark. In these experiments, a spherical particle is released with zero velocity in a fluid initially at rest. The particle accelerates until it reaches its terminal velocity. The latter is controlled by two non-dimensional numbers: the density ratio ρ_p/ρ_f and the Galileo number $Ga = \sqrt{(\rho_p/\rho_f - 1)g_v D^3/\nu}$.

For our comparison, we match case 2 from Mordant and Pinton [26] for which $\rho_p/\rho_f = 2.56$ and $Ga = 255.35$. The simulations are carried out in a domain with dimensions $L_x = L_z = 8D$, and $L_y = 40D$ in the direction of gravity. Three filter sizes are considered: $\delta_f = D/2$, $\delta_f = D/3$, and $\delta_f = D/4$. With the requirement $\delta_f/\Delta x = 4$, this yields 8, 12, and 16 grid points across the particle diameter, respectively. In all these runs, $CFL_{\max} \sim 0.35$.

Fig. 18 shows comparison of the particle velocity time series with the experimentally obtained data by Mordant and Pinton [26]. As can be seen from the figure, the case with resolution $\delta_f/D = 1/4$ shows excellent agreement with the experiment. The difference between the computed terminal velocity and the one reported by Mordant and Pinton [26] is less than 0.3%. The Reynolds number and Froude number based on the terminal velocity are $Re = v_{p,t} d_p/\nu = 368.39$ and $Fr = v_{p,t}/\sqrt{gD} = 1.80$, respectively, in the simulations and $Re = 367.41$ and $Fr = 1.80$ in the experiments. There is good agreement even for the case with resolution $\delta_f/D = 1/3$ where the relative error for the settling velocity is about 2.5%. In the coarsest case $\delta_f/D = 1/2$, the relative error increases to about 13.8%.

6. Conclusion

In this paper, we have presented a novel immersed boundary method, called VFIB, derived by applying the volume-filtering technique of Anderson and Jackson [2]. Without assuming any discretization, we obtain filtered transport equations where the effect of the immersed boundary appears as a forcing on the right-hand side of these equations. This new framework can be regarded as a generalization of previous immersed boundary methods, and provides a theoretical footing for further extensions. We provided extensive details on how this method can be implemented in existing flow solvers and showed that it yields excellent results in two- and three-dimensional cases with static, forcibly moving, and freely moving immersed boundaries.

We shall emphasize that there are no restrictions on the type of computations that can be performed with the VFIB method. As shown in the numerical examples, any moving or static can be accurately represented and accounted for including for problems in fluid-structure interaction, impeller/turbine setups, and Particle-Resolved DNS of particle-laden flows.

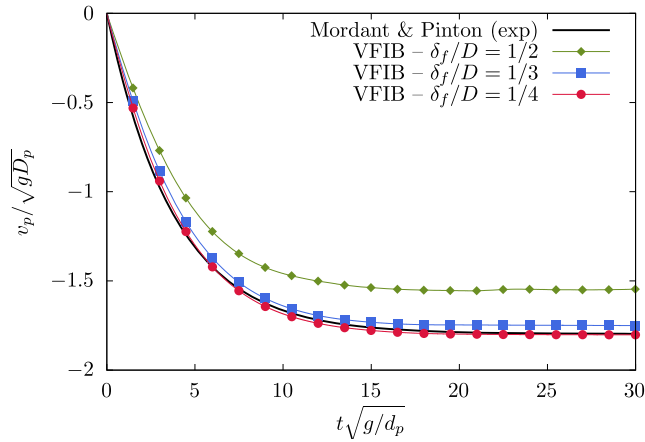


Fig. 18. Time series of the particle settling velocity for the case with $\rho_p/\rho_f = 2.56$ and $Ga = 41798$. The solid line is obtained using the experimental correlation of Mordant and Pinton [26]. The Reynolds number and Froude number based on the terminal velocity are $Re = 367$ and $Fr = 1.80$, respectively.

The present work contains key innovations that enable us to answer several open questions in the literature of immersed boundary methods. First, by volume-filtering the original Navier-Stokes equations, we derived *analytically* the immersed boundary forcing term. This is in contrast to the majority of prior methods that build the IB forcing based on numerical considerations, such as requiring that cells coinciding with the IB have velocities that match the IB velocity at the next step [44,19,38,36,23,7,17,42]. A major aspect of our method is the fact that we intentionally take into account the finite width of the filter. This allows us to carry out the analytical derivation, find an expression for the IB forcing, derive equations for the fluid/solid volume fraction, and forces exerted on the IB by the external flow.

Second, using the VFIB method, we are able to elucidate the role of the internal flow obtained when the IB forcing is limited to the solid-fluid boundary only. As we show in the one-phase formulation, this flow has physical meaning and exists because the immersed object is in reality hollow, has infinitely thin shell, and is filled with identical fluid to the one on the outside. It follows that when the immersed object moves, the fluid inside is also affected, resulting in additional stresses on the boundary. In order to isolate the hydrodynamic force due to the external fluid only, the contribution due to the internal fluid must be removed. The details are given in §2.4. This approach removes the need for ad-hoc fixes, like artificially retracting the immersed boundary, to get accurate hydrodynamic forces.

Third, we showed that the volumes associated with Lagrangian markers depend on the local topology of the interface and can be determined simply using the smearing length ℓ and a triangular tessellation of the interface ($\Delta V_m = \ell(\mathbf{x}_m)A_m$). These Lagrangian volumes arise after discretization of the surface integrals accounting for the IB forcing. At a point $\mathbf{x}_m$ on the interface, the smearing length is the inverse of the surface density at that point $\ell(\mathbf{x}_m) = \Sigma(\mathbf{x}_m)^{-1}$. Its value depends on the local curvature, and choice of filter kernel, and is generally on the order of the filter width δ_f .

Fourth, we provided an efficient procedure to compute the solid volume fractions α_s . The approach is based on solving a Poisson equation for α_s , rather than using the cumbersome and computationally expensive definition. Computing the volume fraction allows efficient tagging of interior/exterior cells, and is used in the procedure to isolate the hydrodynamic stresses due to the external fluid from the total stresses due to both internal and external fluids.

Fifth, we showed a path forward to extend the VFIB method to Large Eddy Simulations with immersed boundaries. By rigorously filtering the Navier-Stokes equations, we showed the presence of sub-filter scale terms. While we have not considered this in the present study, closures for these terms can be carried out using existing LES models, or derived by applying coarse filters to highly-resolved simulations with the VFIB method.

Lastly, we found that the choice of filter kernel does not impact the solution considerably, provided that the immersed boundary is well resolved. Among the four kernels we have considered, the triangle kernel is the simplest to implement and is, thus, the preferred one. The resolution of the immersed boundary depends on the ratio δ_f/δ_c where δ_c is the characteristic corrugation scale of the interface. In two-dimensional cases of flow past a cylinder of diameter D , we found little difference between the filter kernels when $\delta_f/D \leq 1/12$. With increasing resolution, drag and lift coefficients converge towards the reference values to within one or two percent. In three dimensional cases of flow past a sphere, we get accurate drag coefficient, separation angle, and recirculation bubble length with a resolution $\delta_f/D = 1/9$.

CRediT authorship contribution statement

Himanshu Dave: Data curation, Formal analysis, Investigation, Software, Validation, Writing – original draft. **Marcus Herrmann:** Conceptualization, Funding acquisition, Writing – review & editing. **M. Housseem Kasbaoui:** Conceptualization, Funding acquisition, Methodology, Software, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The authors acknowledge support from the US National Science Foundation (award #2028617, CBET-FD). Computing resources were provided by ACCESS allocation PHY200082 and Research Computing at Arizona State University.

Appendix A. Alternative form of the volume-filtered equations

Here, we explain how equations (20) and (21) can be further transformed to get the familiar form used in multiphase flows. Starting from equation (21), we expand the first term on the right-hand side to obtain

$$\rho_f \left(\frac{\partial}{\partial t} (\alpha_f \bar{\mathbf{u}}_f) + \nabla \cdot (\alpha_f \bar{\mathbf{u}}_f \bar{\mathbf{u}}_f) \right) = -\alpha_f \nabla \bar{p} + \alpha_f \nabla \cdot \left(\mu_f \left(\nabla \bar{\mathbf{u}}_f + \nabla \bar{\mathbf{u}}_f^T - \frac{2}{3} (\nabla \cdot \bar{\mathbf{u}}_f) \mathbf{I} \right) + \mathbf{R}_{\mu,f} \right) - \tilde{\mathbf{F}}_{I,f} - \nabla \cdot (\alpha_f \boldsymbol{\tau}_{\text{sf},f}). \quad (\text{A.1})$$

In this usual form, the term $\tilde{\mathbf{F}}_{I,f}$ is given by

$$\tilde{\mathbf{F}}_{I,f} = \mathbf{F}_{I,f} - \nabla \alpha_f \cdot \bar{\boldsymbol{\tau}}, \quad (\text{A.2})$$

where $\bar{\boldsymbol{\tau}} = -\bar{p} \mathbf{I} + \mu_f (\nabla \bar{\mathbf{u}}_f + \nabla \bar{\mathbf{u}}_f^T - 2/3 (\nabla \cdot \bar{\mathbf{u}}_f) \mathbf{I} + \mathbf{R}_{\mu})$ is the filtered stress tensor. Inserting $\bar{\boldsymbol{\tau}}$ in identity (12), we can show

$$\nabla \alpha_f \cdot \bar{\boldsymbol{\tau}} - \iint_{\mathbf{y} \in S_I} \mathbf{n} \cdot \bar{\boldsymbol{\tau}} g(\mathbf{x} - \mathbf{y}) dS = \alpha_f \left(\overline{\nabla \cdot \bar{\boldsymbol{\tau}}} - \nabla \cdot \bar{\boldsymbol{\tau}} \right) = O(\delta_f^2). \quad (\text{A.3})$$

Equation (A.3) combined with the fact that $\bar{\boldsymbol{\tau}} = \bar{\boldsymbol{\tau}} + O(\delta_f^2)$ leads to

$$\tilde{\mathbf{F}}_{I,f} = \iint_{\mathbf{y} \in S_I} \mathbf{n} \cdot (\boldsymbol{\tau} - \bar{\boldsymbol{\tau}}) g(\mathbf{x} - \mathbf{y}) dS + O(\delta_f^2). \quad (\text{A.4})$$

The momentum equation is then written in the following form

$$\rho_f \left(\frac{\partial}{\partial t} (\alpha_f \bar{\mathbf{u}}_f) + \nabla \cdot (\alpha_f \bar{\mathbf{u}}_f \bar{\mathbf{u}}_f) \right) = -\alpha_f \nabla \bar{p} + \alpha_f \nabla \cdot \left(\mu_f \left(\nabla \bar{\mathbf{u}}_f + \nabla \bar{\mathbf{u}}_f^T - \frac{2}{3} (\nabla \cdot \bar{\mathbf{u}}_f) \mathbf{I} \right) + \mathbf{R}_{\mu,f} \right) - \iint_{\mathbf{y} \in S_I} \mathbf{n} \cdot (\boldsymbol{\tau} - \bar{\boldsymbol{\tau}}) g(\mathbf{x} - \mathbf{y}) dS - \nabla \cdot (\alpha_f \boldsymbol{\tau}_{\text{sf},f}). \quad (\text{A.5})$$

References

- [1] Knut Akselvoll, Parviz Moin, Large-eddy simulation of turbulent confined coannular jets, *J. Fluid Mech.* 315 (May 1996) 387–411.
- [2] T.B. Anderson, Roy Jackson, Fluid mechanical description of fluidized beds. Equations of motion, *Ind. Eng. Chem. Fundam.* 6 (4) (November 1967) 527–539.
- [3] T.B. Anderson, Roy Jackson, Fluid mechanical description of fluidized beds. Stability of state of uniform fluidization, *Ind. Eng. Chem. Fundam.* 7 (1) (February 1968) 12–21.
- [4] Sourabh V. Apte, Mathieu Martin, Neelesh A. Patankar, A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows, *J. Comput. Phys.* 228 (8) (May 2009) 2712–2738.
- [5] J. Bardino, J. Ferziger, W. Reynolds, Improved turbulence models based on large eddy simulation of homogeneous, incompressible, turbulent flows, 1983, undefined.
- [6] Joseph Boussinesq, Essai Sur La Théorie Des Eaux Courantes, Mémoires présentés par divers savants à l'Académie des Sciences XXIII (1) (1877) 1–680.
- [7] Wim-Paul Breugem, A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows, *J. Comput. Phys.* (ISSN 0021-9991) 231 (13) (May 2012) 4469–4498, <https://doi.org/10.1016/j.jcp.2012.02.026>.
- [8] Haecheon Choi, Parviz Moin, Effects of the computational time step on numerical solutions of turbulent flow, *J. Comput. Phys.* 113 (1) (July 1994) 1–4.
- [9] Pedro Costa, Bendiks Jan Boersma, Jerry Westerweel, Wim-Paul Breugem, Collision model for fully resolved simulations of flows laden with finite-size particles, *Phys. Rev. E* 92 (5) (November 2015) 053012, <https://doi.org/10.1103/PhysRevE.92.053012>.
- [10] Olivier Desjardins, Guillaume Blanquart, Guillaume Balarac, Heinz Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, *J. Comput. Phys.* 227 (15) (July 2008) 7125–7159.

- [11] Massimo Germano, Ugo Piomelli, Parviz Moin, William H. Cabot, A dynamic subgrid-scale eddy viscosity model, *Phys. Fluids A, Fluid Dyn.* 3 (7) (July 1991) 1760–1765.
- [12] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, *Int. J. Multiph. Flow* 25 (5) (August 1999) 755–794.
- [13] Kai Höfler, Stefan Schwarzer, Navier-Stokes simulation with constraint forces: finite-difference method for particle-laden flows and complex geometries, *Phys. Rev. E* 61 (6) (June 2000) 7146–7160, <https://doi.org/10.1103/PhysRevE.61.7146>.
- [14] Roy Jackson, *The Dynamics of Fluidized Particles*, Cambridge University Press, ISBN 978-0-521-78122-0, September 2000.
- [15] T.A. Johnson, V.C. Patel, Flow past a sphere up to a Reynolds number of 300, *J. Fluid Mech.* 378 (January 1999) 19–70.
- [16] M.H. Kasbaoui, T. Kulkarni, F. Bisetti, Direct numerical simulations of the swirling von Kármán flow using a semi-implicit moving immersed boundary method, *Comput. Fluids* (ISSN 0045-7930) 230 (November 2021) 105132, <https://doi.org/10.1016/j.compfluid.2021.105132>.
- [17] Tobias Kempe, Jochen Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, *J. Comput. Phys.* (ISSN 0021-9991) 231 (9) (May 2012) 3663–3684, <https://doi.org/10.1016/j.jcp.2012.01.021>.
- [18] Ming-Chih Lai, Charles S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2) (May 2000) 705–719.
- [19] Jongho Lee, Jungwoo Kim, Haecheon Choi, Kyung-Soo Yang, Sources of spurious force oscillations from an immersed boundary method for moving-body problems, *J. Comput. Phys.* (ISSN 0021-9991) 230 (7) (April 2011) 2677–2695, <https://doi.org/10.1016/j.jcp.2011.01.004>.
- [20] D.K. Lilly, A proposed modification of the Germano subgrid-scale closure method, *Phys. Fluids A, Fluid Dyn.* 4 (3) (March 1992) 633–635.
- [21] C. Liu, X. Zheng, C.H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.* 139 (1) (January 1998) 35–57.
- [22] X.Y. Lu, C. Dalton, Calculation of the timing of vortex formation from an oscillating cylinder, *J. Fluids Struct.* 10 (5) (July 1996) 527–541.
- [23] Kun Luo, Zeli Wang, Jianren Fan, Kefa Cen, Full-scale solutions to particle-laden flows: multidirect forcing and immersed boundary method, *Phys. Rev. E* 76 (6) (December 2007) 066709.
- [24] R. Mittal, H. Dong, M. Bozkurtas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* 227 (10) (May 2008) 4825–4852.
- [25] W.C. Moore, S. Balachandar, Lagrangian investigation of pseudo-turbulence in multiphase flow using superposable wakes, *Phys. Rev. Fluids* 4 (11) (November 2019) 114301, <https://doi.org/10.1103/PhysRevFluids.4.114301>.
- [26] N. Mordant, J.-F. Pinton, Velocity measurement of a settling sphere, *Eur. Phys. J. B, Condens. Matter Complex Syst.* 18 (2) (November 2000) 343–352.
- [27] C.S. Peskin, The fluid dynamics of heart valves: experimental, theoretical, and computational methods, *Annu. Rev. Fluid Mech.* 14 (1) (1982) 235–259, <https://doi.org/10.1146/annurev.fl.14.010182.001315>.
- [28] Charles S. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (2) (October 1972) 252–271.
- [29] Charles S. Peskin, The immersed boundary method, *Acta Numer.* 11 (January 2002) 479–517, <https://doi.org/10.1017/S0962492902000077> (ISSNs 1474-0508, 0962-4929).
- [30] Charles D. Pierce, Parviz Moin, Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion, *J. Fluid Mech.* 504 (April 2004) 73–97.
- [31] H.R. Pruppacher, B.P. Le Clair, A.E. Hamielec, Some relations between drag and flow pattern of viscous flow past a sphere and a cylinder at low and intermediate Reynolds numbers, *J. Fluid Mech.* 44 (4) (December 1970) 781–790, <https://doi.org/10.1017/S0022112070002148> (ISSNs 1469-7645, 0022-1120).
- [32] Alexandre M. Roma, Charles S. Peskin, Marsha J. Berger, An adaptive version of the immersed boundary method, *J. Comput. Phys.* (ISSN 0021-9991) 153 (2) (August 1999) 509–534, <https://doi.org/10.1006/jcph.1999.6293>.
- [33] Frederick W. Roos, William W. Willmarth, Some experimental results on sphere and disk drag, *AIAA J.* (ISSN 0001-1452) 9 (2) (February 1971) 285–291, <https://doi.org/10.2514/3.6164>.
- [34] M. Schäfer, S. Turek, F. Durst, E. Krause, R. Rannacher, Benchmark computations of laminar flow around a cylinder, in: Ernst Heinrich Hirschel (Ed.), *Flow Simulation with High-Performance Computers II: DFG Priority Research Programme Results 1993–1995*, in: Notes on Numerical Fluid Mechanics (NNFM), Vieweg+Teubner Verlag, Wiesbaden, ISBN 978-3-322-89849-4, 1996, pp. 547–566.
- [35] L. Schiller, A drag coefficient correlation, *Z. Ver. Dtsch. Ing.* 77 (1933) 318–320.
- [36] Lennart Schneiders, Daniel Hartmann, Matthias Meinke, Wolfgang Schröder, An accurate moving boundary formulation in cut-cell methods, *J. Comput. Phys.* 235 (February 2013) 786–809.
- [37] O. Sen, N.J. Gaul, S. Davis, K.K. Choi, G. Jacobs, H.S. Udaykumar, Role of pseudo-turbulent stresses in shocked particle clouds and construction of surrogate models for closure, *Shock Waves* (ISSN 1432-2153) 28 (3) (May 2018) 579–597, <https://doi.org/10.1007/s00193-017-0801-1>.
- [38] Jung Hee Seo, Rajat Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, *J. Comput. Phys.* (ISSN 0021-9991) 230 (19) (August 2011) 7347–7363, <https://doi.org/10.1016/j.jcp.2011.06.003>.
- [39] Nitin Sharma, Neelesh A. Patankar, A fast computation technique for the direct numerical simulation of rigid particulate flows, *J. Comput. Phys.* 205 (2) (May 2005) 439–457.
- [40] Sadatoshi Taneda, Experimental investigation of the wake behind a sphere at low Reynolds numbers, *J. Phys. Soc. Jpn.* (ISSN 0031-9015) 11 (10) (October 1956) 1104–1108, <https://doi.org/10.1143/JPSJ.11.1104>.
- [41] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, *J. Fluid Mech.* 6 (4) (November 1959) 547–567.
- [42] Silvio Tschisgale, Tobias Kempe, Jochen Fröhlich, A non-iterative immersed boundary method for spherical particles of arbitrary density ratio, *J. Comput. Phys.* 339 (June 2017) 432–452.
- [43] M. Uhlmann, *First Experiments with the Simulation of Particulate Flows*, CIEMAT (Technical Report No. 1020), July 2003.
- [44] Markus Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2) (November 2005) 448–476.
- [45] Salih Ozen Unverdi, Grétar Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* (ISSN 0021-9991) 100 (1) (May 1992) 25–37, [https://doi.org/10.1016/0021-9991\(92\)90307-K](https://doi.org/10.1016/0021-9991(92)90307-K).
- [46] Stephen Whitaker, Flow in porous media I: a theoretical derivation of Darcy's law, *Transp. Porous Media* 1 (1) (March 1986) 3–25.
- [47] Xiaolei Yang, Xing Zhang, Zhilin Li, Guo-Wei He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, *J. Comput. Phys.* (ISSN 0021-9991) 228 (20) (November 2009) 7821–7836, <https://doi.org/10.1016/j.jcp.2009.07.023>.
- [48] M.M. Zdravkovich, Smoke observations of the formation of a Kármán vortex street, *J. Fluid Mech.* 37 (3) (July 1969) 491–496.
- [49] Kun Zhou, S. Balachandar, An of the spatio-temporal resolution of the immersed boundary method with direct forcing, *J. Comput. Phys.* (ISSN 0021-9991) 424 (January 2021) 109862, <https://doi.org/10.1016/j.jcp.2020.109862>.