An immersed boundary method based on domain decomposition

Krishnan, Navaneetha; Viré, Axelle; Schmehl, Roland; van Bussel, Gerard

DOI
10.1016/j.compfluid.2020.104500

Publication date
2020

Document Version
Final published version

Published in
Computers and Fluids

Computers and Fluids 202 (2020) 104500

Contents lists available at ScienceDirect

Computers and Fluids

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

An immersed boundary method based on domain decomposition

Navaneetha Krishnan, Axelle Viré*, Roland Schmehl, Gerard van Bussel

Wind Energy Section, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, the Netherlands

ARTICLE INFO

Article history:
Received 21 February 2019
Revised 6 January 2020
Accepted 6 March 2020
Available online 12 March 2020

Keywords:
Immersed boundary method
Finite element method
Fluid-structure interactions

ABSTRACT

A novel immersed boundary method based on a domain decomposition approach is proposed in the context of a finite element discretisation method. It is applicable to incompressible flows past rigid, deforming, or moving bodies. In this method, unlike most immersed boundary methods, strong boundary conditions are imposed in the regions of the computational domain that are occupied by the structure. In order to achieve this, the proposed formulation decomposes the computational domain by splitting the finite element test functions into solid and fluid parts. In the continuous Galerkin formulation, this produces a smeared representation of the fluid-structure interface. The absence of an immersed boundary forcing term implies that the method itself has no influence on the CFL stability criterion. Furthermore, the stiffness matrix in the momentum equation is sparser than compared with other forcing immersed boundary methods, and symmetry and positive-definiteness of the Laplacian operator in the pressure equation is preserved. As shown in this paper, stability and accurate imposition of boundary conditions make the method promising for high Reynolds number flows. The method is applied to the simulations of two-dimensional laminar flow over stationary and moving cylinders, as well as a moderately high Reynolds number flow past an aerofoil. Good results are obtained when compared with those from previous experimental and numerical studies.

1. Introduction

Fluid-structure interactions (FSI) are encountered in many engineering applications, such as the flow past large wind turbine blades, the deployment of spacecraft parachutes, or blood flows through arteries. Immersed boundary methods are attractive to numerically simulate these problems, especially when the geometry of the structure is complex and arbitrarily deforming [1–3]. These methods immerse the structure in a fluid domain and mimic its effect on the fluid dynamics. This contrasts with body-conforming methods, in which the fluid domain excludes the structure and the governing equations of the flow field are solved only in regions surrounding the structure. The original immersed boundary method was introduced by Peskin [4] to simulate cardiac flows. Since then, several variations of this approach have been proposed, as detailed hereafter.

Based on how the fluid-structure interface is represented on the discrete mesh, immersed boundary methods can be classified as sharp-interface and diffused-interface methods. In diffused-interface methods, the discrete representation of the fluid-structure interface spreads over one or more grid cells. Therefore, the interface is smeared and differs from the actual surface boundary of the structure. Methods mimicking the effect of the structure through an additional forcing term in the fluid's equations of motion often fall under this category. The simplest of the forcing methods are the penalty methods [5–7]. In that case, the force term that is added to the right-hand-side of the fluid's momentum equation is proportional to the difference between fluid and structure velocities with a certain penalty factor. The latter, which is difficult to interpret physically, can change the strength at which the penalty condition is enforced. Such penalised forcing methods are also popular in the context of finite element methods [8–11]. The penalty force is always masked by a stepping function to limit the region in which it acts. The stepping function can be a Dirac function across the interface element [6,8], weighted functions spread across multiple elements to avoid numerical oscillations [7] or a conservatively projected solid concentration field that marks the presence of the body [11]. A slightly modified version of the penalty method is the feedback control method [12]. In this method, in addition to the damping term, the forcing term also has a stiffness term. From a physical point of view, the boundary condition is then imposed as a damped oscillator [1]. A difference between the penalty forcing method and the feedback control method lies in the way the boundary condition is satisfied: it is always lagging behind in the penalty forcing method, while it is oscillatory in the feedback control method. Another common

* Corresponding author.
E-mail address: a.c.vire@tudelft.nl (A. Viré).

https://doi.org/10.1016/j.compfluid.2020.104500
0045-7930/© 2020 Elsevier Ltd. All rights reserved.

forcing method is the direct forcing method, in which the additional forcing term balances out the convective, viscous, and pressure gradient terms in the Navier–Stokes equations [13,14]. In this class of immersed boundary methods, the physical interpretation of the forcing term is clearer than in the penalty method. A drawback, however, is that the implicit evaluation of the forcing term leads to a denser stiffness matrix in the discrete system of equations. In order to solve this, variations of the original direct forcing method have been proposed and evaluate the forcing term at different time levels. For example, a semi-implicit staggered approach was proposed and computed an intermediate velocity field without the forcing term, then calculated the forcing term using the intermediate velocity field, and eventually made a velocity correction using the forcing term [15,16]. Implicit approaches are usually preferred in terms of accuracy, since the boundary conditions are imposed while solving the momentum equations. However, in all their variants, the no-slip condition at the interface could be violated in the staggered step to project velocity onto a solenoidal field. Also, solution to the Poisson's system that has to be solved for the projection is affected by the quality of the grid, both inside and outside the immersed boundary. This can be particularly challenging for problems involving thin membranes.

One of the undesirable features of diffused-interface methods is that the discrete fluid-structure interface is smeared across several mesh nodes. Cut-cell methods are a type of sharp-interface methods that eliminate this issue by locally reconstructing the mesh to align it with the boundary [17–20]. However, this approach is mostly applied to two-dimensional problems as the mesh reconstruction in three-dimensional cases is inherently difficult. Even in two-dimensional problems, it was shown that a major challenge is to properly define and manage the topology of cut cells [3]. Another sharp-interface approach is the hybrid Cartesian immersed boundary method, which uses a Cartesian grid and applies strong boundary conditions at the nodes located in the structure [21]. This method can be more accurate in applying the fluid-structure interface conditions than the previous immersed boundary methods. However, the use of a Cartesian grid is inefficient to resolve boundary layers near complex geometries, hence limiting the applicability of the method to flows past simple geometries. The curvilinear immersed boundary method [22] is an extension of this method and uses curvilinear grids. The latter are better than Cartesian grids to resolve boundary layers for relatively simple curved geometries.

Finally, a completely different approach, relevant for both sharp and diffused interfaces, is based on the idea that the forcing term can be viewed as a Lagrange multiplier that imposes a velocity boundary condition in regions occupied by the immersed body [23–25]. The role of the multiplier is then similar to the role of the pressure field in an incompressible flow, where pressure is effectively used to satisfy the divergence-free condition. This method presents several advantages as it is able to impose boundary conditions accurately, while preserving the symmetry of the Poisson's system. Also, the operators can be tweaked to impose sharp-interface or diffused-interface conditions.

In this paper, a novel method is proposed to accurately impose the boundary condition at the fluid-structure interface, without limitations to Cartesian grids and without requiring additional equations to be solved. The method is developed in the context of the finite-element discretisation and applies the no-slip condition at the interface as a strong boundary condition in the fluid's momentum equations. This is achieved by decomposing the computational domain into two distinct regions, one for the fluid and another one for the structure. The domain decomposition is performed by splitting the finite element test functions into two parts. The momentum equation is solved for across the whole computational domain with a strong imposition of the boundary condition inside the solid phase. The Poisson's equation for pressure is solved on the computational sub-domain that belongs to the fluid phase. The paper is organised as follows. Section 2 outlines the basic principle of the method and presents the governing equations in the solid and fluid sub-domains. Section 3 converts the continuous equations into discrete forms. Section 4 demonstrates the performance of the method on several test cases. First, three laminar cases are considered: Taylor-Couette flow, flow past a stationary two-dimensional cylinder, and flow past an oscillating cylinder. Second, flow past a NACA0012 aerofoil at a Reynolds number of $Re = 1000$ is simulated to evaluate the stability and accuracy of the proposed method at a moderately high Reynolds number. Results are compared against analytical solutions, experimental data and other numerical simulations. In both flow regimes, results show good convergence rates and excellent agreement with existing studies.

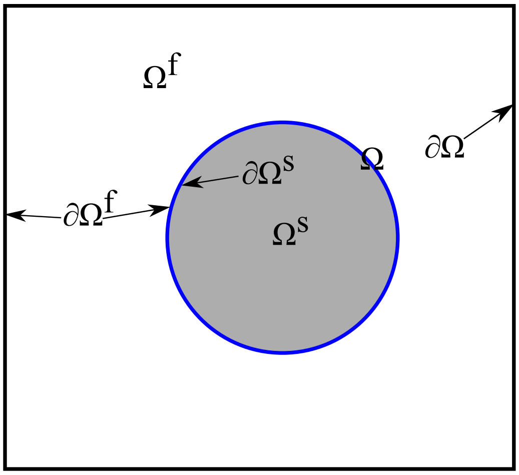


Fig. 1. Schematic representation of the computational domain (Ω), and solid (Ω^s) and fluid (Ω^f) sub-domains. $\partial\Omega$ is the boundary of the computational domain. $\partial\Omega^f$ and $\partial\Omega^s$ are the boundaries of the fluid and solid sub-domains. The blue curve is the immersed boundary that is shared between the solid and fluid sub-domains. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2. Continuous formulation

2.1. Strong form of the governing equations

As stated above, the computational domain in immersed boundary problems encompasses both fluid and solid regions. This is illustrated by Fig. 1, which shows how the computational domain Ω is decomposed into two sub-domains: a fluid domain (Ω^f) and a solid domain (Ω^s), with $\Omega = \Omega^f \cup \Omega^s$ and $\Omega^f \cap \Omega^s = 0$. The boundary of the computational domain is denoted by $\partial\Omega$, while boundaries of the fluid and solid sub-domains are $\partial\Omega^f$ and $\partial\Omega^s$, respectively. Fig. 1 also shows that the immersed boundary of the solid domain ($\partial\Omega^s$ shown by the blue curve) is a sub-set of the boundary of the fluid domain, i.e. $\partial\Omega^f = \partial\Omega \cup \partial\Omega^s$. The boundary $\partial\Omega$ can be further decomposed into $\partial\Omega^D$, $\partial\Omega^N$, and $\partial\Omega^R$ to apply Dirichlet, Neumann, and Robin boundary conditions, respectively.

In the fluid domain, the advection-diffusion equation that governs the time evolution of a scalar field c is given in the conservative form by,

$$\int_{\Omega^f}\left(\frac{\partial c}{\partial t} + \nabla\cdot(\mathbf{u}c) - \nabla\cdot(\bar{\bar{k}}\nabla c) - F\right)dV = 0, \tag{1}$$

in which $\mathbf{u}$ is the transport velocity vector, $\bar{\bar{k}}$ is the diffusivity tensor, and F represents any forcing terms. The scalar field c is defined on the whole computational domain and can be, for example, one of the velocity components, a species concentration, the temperature, the turbulent kinetic energy k, the turbulent frequency ω, or the turbulent dissipation ϵ. The density ρ of the fluid is assumed to be constant, so that the continuity equation for velocity is given by,

$$\int_{\Omega^f} (\nabla \cdot \mathbf{u}) dV = 0. \tag{2}$$

In the solid domain Ω^s, the transport variable c is set to the structural value c^s, which generally speaking is either prescribed or obtained from the resolution of the structural dynamics equations, such that

$$\int_{\Omega^s} \frac{\partial c}{\partial t} dV = \int_{\Omega^s} \frac{\partial c^s}{\partial t} dV. \tag{3}$$

In this paper, results will be shown for fixed structures or structures undergoing a prescribed motion.

Eqs. (1) –(3) are the governing equations of the problem with the following boundary conditions in space and time.

$$c(\cdot, 0) = c_0(\cdot) \quad \text{in } \Omega^f, \tag{4a}$$

$$c(\cdot, 0) = c_0^s(\cdot) \quad \text{in } \Omega^s, \tag{4b}$$

$$c = \tilde{c} \quad \text{in } \partial\Omega^D, \tag{4c}$$

$$\bar{\bar{\kappa}}\nabla c \cdot n = q \quad \text{in } \partial\Omega^N, \tag{4d}$$

$$-\bar{\bar{\kappa}}\nabla c \cdot n = h(c - c_a) \quad \text{in } \partial\Omega^R, \tag{4e}$$

where, $\tilde{c}$ is the fixed condition applied on $\partial\Omega^D$, q is the flux across $\partial\Omega^N$, h is the surface transfer coefficient on $\partial\Omega^R$, and c_a is an ambient field value.

2.2. Weak form of the governing equations

The weak form of Eq. (1) is obtained by multiplying its left-hand-side with a test function ψ^f and using Green's first identity to integrate by parts over the domain Ω^f, such that,

$$\int_{\Omega^f} \left(\psi^f \frac{\partial c}{\partial t} - \nabla\psi^f \cdot \mathbf{u}c + \nabla\psi^f \cdot \bar{\bar{\kappa}} \cdot \nabla c - \psi^f F \right) dV + \int_{\partial\Omega^f} \left(\psi^f (\mathbf{u}c - \bar{\bar{\kappa}}\nabla c) \cdot \mathbf{n} \right) dS = 0, \tag{5}$$

where $\mathbf{n}$ is the surface normal vector. Note that $\partial\Omega^f$ includes both the boundaries of the computational domain (Ω) and the boundary between the fluid domain (Ω^f) and the solid domain (Ω^s). Additionally, the integral over the parts of $\partial\Omega^f$ that coincide with the boundaries of the computational domain (Ω) is often subdivided into various components depending on the boundary conditions that are applied. For simplicity it is assumed that the parts of $\partial\Omega^f$ coinciding with the boundaries $\partial\Omega$ are closed (with $\mathbf{u} \cdot \mathbf{n} = 0$) and that $\nabla c \cdot \mathbf{n} = 0$ (Neumann condition). Furthermore, the test function (ψ^f) is constructed such that its value is zero at the fluid-solid interface (blue curve in Fig. 1). Under these conditions, the boundary integral in Eq. (5) cancels out, which yields

$$\int_{\Omega^f} \left(\psi^f \frac{\partial c}{\partial t} - \nabla\psi^f \cdot \mathbf{u}c + \nabla\psi^f \cdot \bar{\bar{\kappa}} \cdot \nabla c - \psi^f F \right) dV = 0. \tag{6}$$

Eq. (6) holds in the region Ω^f covered by the fluid. In order to write the equations of motion of the transport variable c in the whole computational domain $\Omega = \Omega^f \cup \Omega^s$, the weak forms of the fluid- and solid-governing equations need to be added together. The weak form of Eq. (3) is obtained by testing it with a test function ψ^s. Adding this to Eq. (6) yields

$$\int_{\Omega} \left((\psi^f + \psi^s) \frac{\partial c}{\partial t} - \nabla\psi^f \cdot \mathbf{u}c + \nabla\psi^f \cdot \bar{\bar{\kappa}} \cdot \nabla c - \psi^f F \right) dV = \int_{\Omega} \left(\psi^s \frac{\partial c^s}{\partial t} \right) dV. \tag{7}$$

The weak form of Eq. (2) is obtained by multiplying the equation with another test function ξ^f, as,

$$\int_{\Omega^f} \xi^f (\nabla \cdot \mathbf{u}) dV = 0. \tag{8}$$

Note that Eq. (8) is not integrated by parts and continuity is only evaluated in Ω^f. This choice will be reasoned in the following section. Eqs. (8) and (7) form the weak formulation associated with Eqs. (1)–(3).

3. Discrete formulation

3.1. General formulation

In this paper, continuous piecewise-linear finite element shape functions (ϕ_i) are used to approximate the continuous flow variables on the discretised domain, so that

$$c \approx \sum_{i=1}^{N} c_i \phi_i, \quad \mathbf{u} \approx \sum_{i=1}^{N} \mathbf{u}_i \phi_i, \quad p \approx \sum_{i=1}^{N} p_i \phi_i, \tag{9}$$

in which N is the number of nodes in the discretised domain and p is the fluid pressure.

Several key points have to be noted before defining the test functions for the problem. First, although $\Omega^f \cap \Omega^s = 0$ in a continuous sense, in the present discrete formulation as indicated in Fig. 2, in which the shaded interface element is part of both Ω^f and Ω^s, $\Omega^f \cap \Omega^s \neq 0$. Because of this overlap, the test functions ψ^f and ψ^s used in the transport equation have to be complements of each other to avoid doubling the common term, $\partial c/\partial t$. Second, the pressure gradient operator will be tested using ψ^f, and thus limited to Ω^f only. Thus, to make the Laplacian operator symmetric, evaluation of continuity, and thereby ξ^f, is also limited to Ω^f. Finally, unlike ψ^f, the function ξ^f does not have a complementary solid part. Thus, to fully evaluate continuity in Ω^f, ξ^f should not be forced to zero at the solid end of the interface. For this reason, Eq. (8) is not integrated by parts since the surface integral term would not vanish based on this definition of ξ^f.

The test functions (ψ_i, ξ_i) are chosen from a function space that is appropriate for the problem [26]. For a continuous Galerkin approach this coincides with the function space from which the shape functions are derived. We define ψ_i^f, ψ_i^s, and ξ_i^f as,

$$\psi_i^f = \phi_i(1 - \alpha_i^{n+1}), \quad \psi_i^s = \phi_i \alpha_i^{n+1}, \tag{10a}$$

$$\exists x \in \alpha_e^{n+1} : x < 1 \Rightarrow \xi_k^f = \phi_k. \tag{10b}$$

Here, α (blue line in Fig. 2) is a solid concentration field that represents the position of the structure on the fluid mesh, and the superscript $n+1$ denotes that the solid position from the forthcoming time step is considered. It is obtained by a consistent interpolation or a bounded conservative projection of a unitary field from the solid mesh onto the fluid mesh [27,28]. Furthermore, the indices e and k in Eq. (10b) iterate over the total number of elements and the local node number in the element, respectively. In this method, the test functions ψ_i^f are the nodal shape functions of the fluid nodes (•, Fig. 2). They have a non-zero value in the shaded region in Fig. 3b. Similarly, the test functions ψ_i^s are the nodal shape functions of the solid nodes (■, Fig. 2) and have

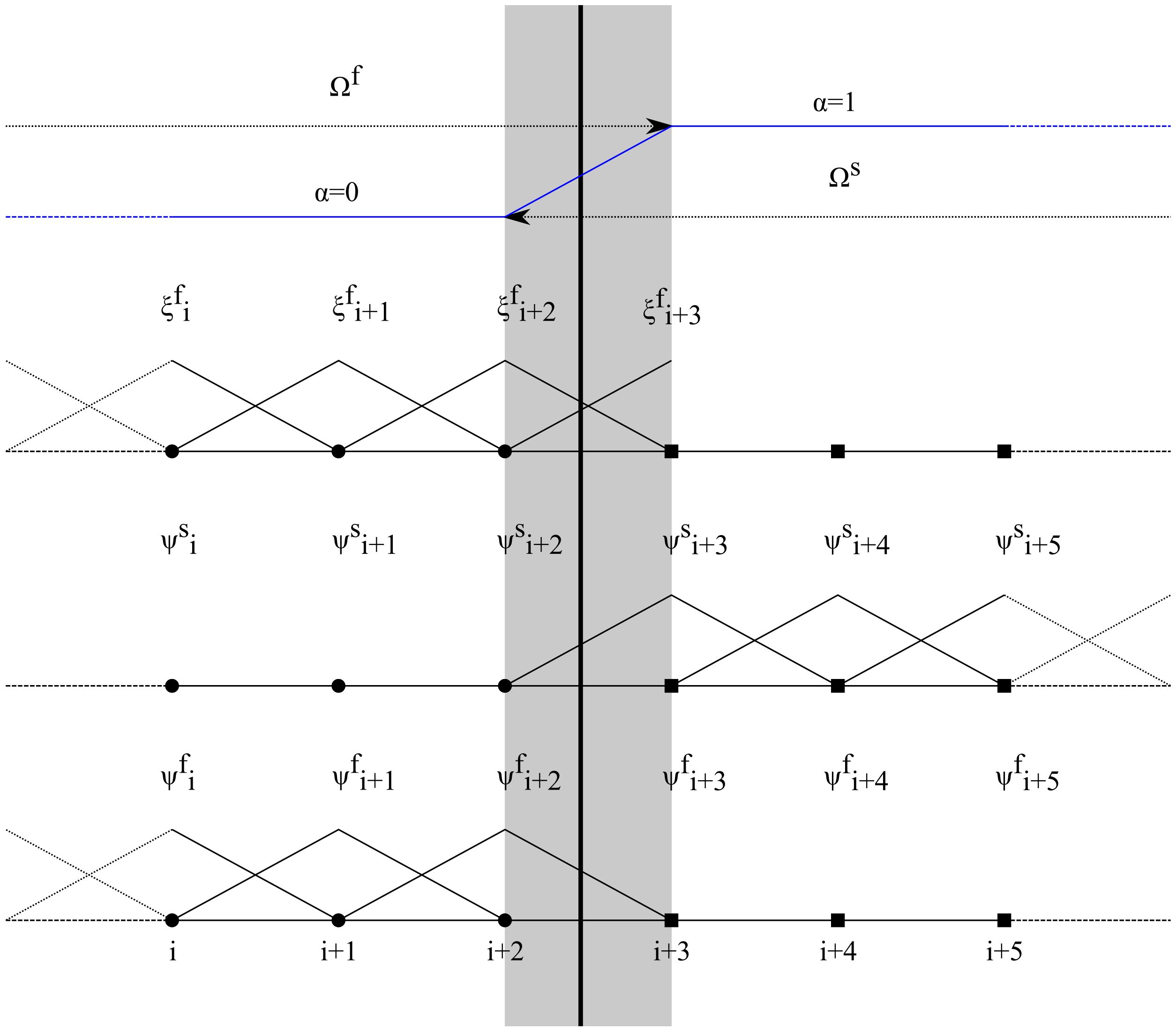


Fig. 2. Schematic representation of the solid (ψ^s), and fluid (ψ^f, ξ^f) linear test functions in a one-dimensional case. The shaded element represents the interface region between the fluid and solid domains. The blue line on top represents the variation of α across the one-dimensional computational domain. ψ^s and ψ^f are used in the transport equations in the solid and fluid domains, respectively. ξ^f is used to enforce the continuity constraint in the fluid domain. Note that ξ^f is discontinuous at the solid end of the interface ($i+3$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a non-zero value in the shaded region in Fig. 3c. Unlike ψ_i^f and ψ_i^s which are defined on the global node basis, the functions ξ_i^f in Eq. (10b) are defined on the local node basis. It is essentially ψ_i^f, without the requirement that its value is zero at the fluid-solid interface. As explained before, the function ξ^f is discontinuous at the solid end of the domain (Fig. 2, node $i+3$).

Eqs. (8) and (7) can be written in discrete acceleration form as,

$$(M+\theta\Delta t(A-D))\frac{c^{n+1}-c^n}{\Delta t} = -(A-D)c^n + f + f^s, \tag{11a}$$

$$\bar{\mathbf{C}}^T\mathbf{u}^{n+1} = 0. \tag{11b}$$

In Eq. (11), M is the mass matrix, A is the advection matrix, and D is the diffusion matrix. Additionally, $\bar{\mathbf{C}}^T$ is the discrete divergence operator (superscript T denotes the transpose). The variable Δt denotes the time step that can vary in the course of the simulation to satisfy a certain CFL value, and θ is a time differencing parameter to choose between different temporal integration schemes ($\theta = 0$ for an explicit scheme, $\theta = 0.5$ for the semi-implicit Crank-Nicolson scheme, and $\theta = 1$ for a fully implicit discretisation). Finally, f^s is the immersed body forcing term and f represents other forcing terms (e.g. gravity force). More specifically, the discrete operators and the forcing terms in the discrete equations are given by,

$$M_{ij} = \int_\Omega (\psi_i^f + \psi_i^s)\phi_j dV, \tag{12a}$$

$$A_{ij} = -\int_\Omega \nabla\psi_i^f \cdot \mathbf{u}^n \phi_j dV, \tag{12b}$$

$$D_{ij} = \int_\Omega \nabla\psi_i^f \cdot \bar{\bar{\kappa}} \cdot \nabla\phi_j dV, \tag{12c}$$

$$\bar{\mathbf{C}}^T_{ij} = \int_\Omega \xi_i^f \nabla\phi_j dV, \tag{12d}$$

$$f_i = \int_\Omega \psi_i^f \phi_j F_j^n dV, \tag{12e}$$

$$f_i^s = \int_\Omega \psi_i^s \phi_j \frac{c_j^{s(n+1)} - c_j^n}{\Delta t} dV. \tag{12f}$$

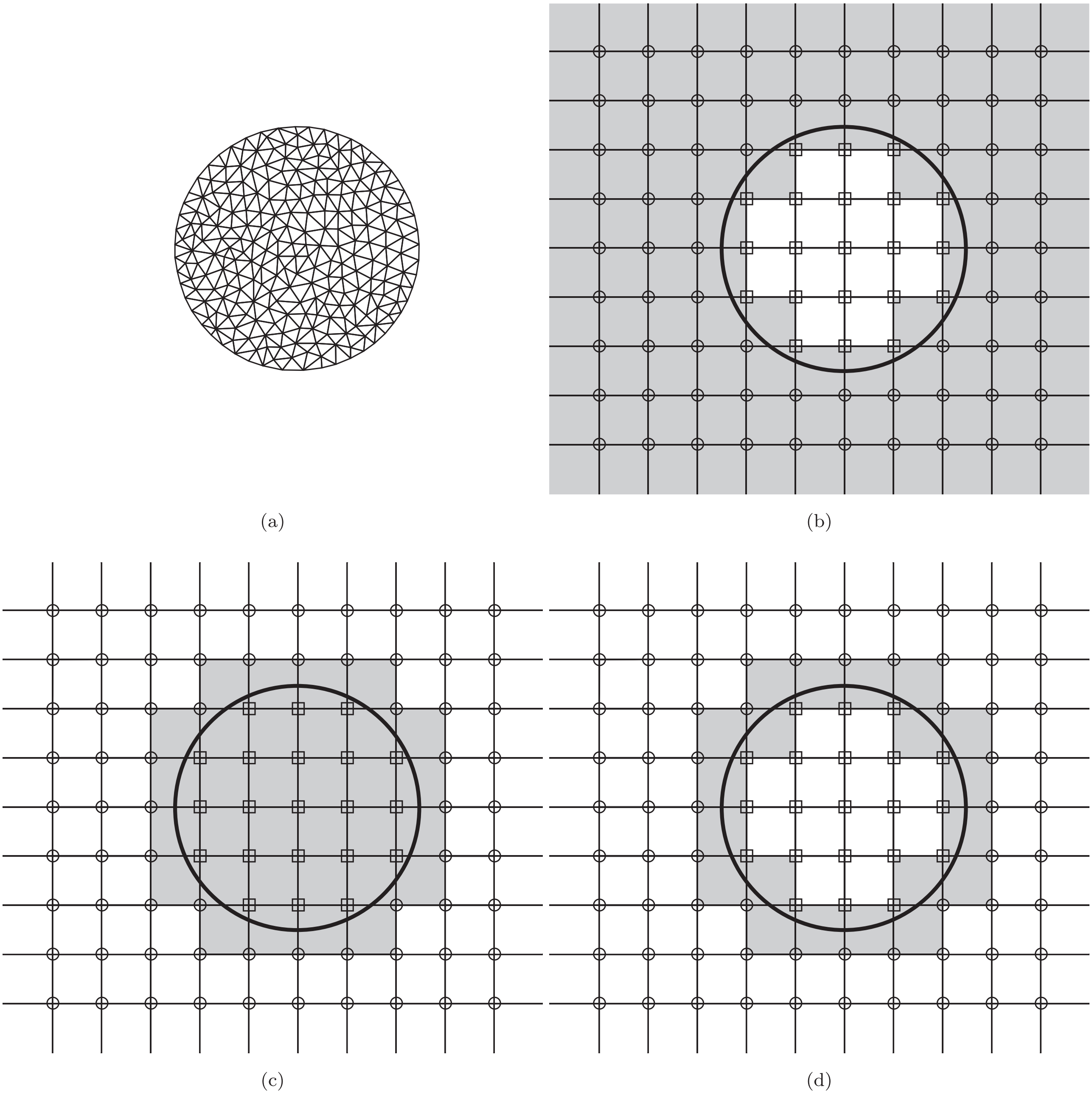


Fig. 3. Schematic representation of the solid (a) and fluid (b, c, d) domains. Figure (a) shows the solid grid (Γ^s). Figures (b, c, d) show the fluid grid (Γ^f), with circular markers ($\circ$) representing fluid nodes ($\alpha^{n+1} = 0$) and square markers ($\Box$) representing solid nodes ($\alpha^{n+1} > 0$), where strong velocity boundary conditions are imposed. Shaded elements represent: (b) the domain in which the fluid operators act, (c) the domain in which the solid operator acts, and (d) the interface region between the solid and fluid domains.

Fig. 3b and c show the decomposed domains in which these operators act. The operator M acts on the whole domain, whilst $A, D, \mathbf{C}^T$, and f acts in the shaded fluid domain in Fig. 3b. The immersed body forcing term, f^s, acts in the shaded solid domain in Fig. 3c.

3.2. Application to the Navier–Stokes equations

The discrete Navier–Stokes momentum equations are obtained from Eq. (11a) by treating velocity as a transport variable and adding $\mathbf{C}p^{n+\theta}$ to the right-hand-side, as

$$(\mathbf{M} + \theta \Delta t(\mathbf{A} - \mathbf{D}))\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -(\mathbf{A} - \mathbf{D})\mathbf{u}^n + \mathbf{C}p^{n+\theta} + \mathbf{f} + \mathbf{f}^s, \quad (13)$$

where,

$$\mathbf{C}_{ij} = -\int_\Omega \psi_i^f \nabla \xi_j^f dV, \quad \mathbf{C}_{ij}^T = \int_\Omega \xi_i^f \nabla \psi_j^f dV. \quad (14)$$

Bold letters denote that the matrices ($\mathbf{M}, \mathbf{A}, \mathbf{D}, \mathbf{f}$, and $\mathbf{f}^s$) are now in block matrix form and $\mathbf{u}^*$ is the predicted velocity. The matrix $-\mathbf{C}$ is the discrete pressure gradient operator. Note that we used ξ^f as pressure basis functions. This is because, just like continuity, pressure is also defined only in Ω^f.

When solving the Navier–Stokes equations, it is common to decouple the solution procedures for velocity and pressure. Chorin's projection method [29] is used here for fractional time stepping. The procedure starts by solving the pressure Poisson's equation to compute an initial pressure field. The Poisson's equation for pressure is obtained by taking the divergence of Eq. (13) and applying the incompressibility condition (Eq. (11b)) on the predicted velocity, i.e.

$$\bar{\mathbf{L}} p^{\theta} = -\bar{\mathbf{C}}^{T}\left(\mathbf{M}^{-1}\mathbf{RHS} + \frac{\mathbf{u}^{0}}{\Delta t}\right) + \cancel{\frac{\bar{\mathbf{C}}^{T}\mathbf{u}^{*}}{\Delta t}}. \tag{15}$$

Here, **RHS** contains all the terms on the right hand side of Eq. (13) except the pressure gradient term ($\mathbf{C}p^{n+\theta}$), and $\bar{\mathbf{L}}$ is a discrete Laplacian expressed as $\bar{\mathbf{L}} = \bar{\mathbf{C}}^{T}\mathbf{M}^{-1}\mathbf{C}$. However, $\bar{\mathbf{L}}$ is not symmetric. A symmetric Poisson's system of equations can be obtained by reformulating the continuity equation for velocity using $\mathbf{C}^{T}$, as

$$\mathbf{C}^{T}\mathbf{u}^{*} = -\mathbf{C}^{T(s)}\mathbf{u}^{*} \equiv -\mathbf{C}^{T(s)}\mathbf{u}^{s(n+1)}, \tag{16a}$$

$$\mathbf{L} p^{\theta} = -\mathbf{C}^{T}\left(\mathbf{M}^{-1}\mathbf{RHS} + \frac{\mathbf{u}^{0}}{\Delta t}\right) - \frac{\mathbf{C}^{T(s)}\mathbf{u}^{s(1)}}{\Delta t}. \tag{16b}$$

Here,

$$\mathbf{C}^{T(s)}_{ij} = \int_{\Omega} \xi_i^{f}\nabla\psi_j^{s} dV \equiv \int_{\Omega} \xi_i^{f}\nabla(\phi_j - \psi_j^{f}) dV \equiv \bar{\mathbf{C}}^{T}_{ij} - \mathbf{C}^{T}_{ij}, \tag{17a}$$

$$\mathbf{L} = \mathbf{C}^{T}\mathbf{M}^{-1}\mathbf{C}. \tag{17b}$$

An intermediate velocity field ($\mathbf{u}^{*}$) is then obtained by solving the momentum equation (Eq. (13)). In general, the predicted velocity field is not divergence-free and will not satisfy Eq. (16a). The projection method tackles this issue by defining an irrotational pressure correction, determined by solving the following Poisson equation,

$$\mathbf{L}\Delta p^{n+\theta} = \frac{-(\mathbf{C}^{T} + \mathbf{C}^{T(s)})(\theta\mathbf{u}^{*} + (1-\theta)\mathbf{u}^{n})}{\Delta t} \equiv \frac{-\bar{\mathbf{C}}^{T}(\theta\mathbf{u}^{*} + (1-\theta)\mathbf{u}^{n})}{\Delta t}, \tag{18}$$

and correcting the velocity field as,

$$\mathbf{u}^{n+1} = \mathbf{u}^{*} + \Delta t\mathbf{M}^{-1}\mathbf{C}\Delta p^{n+\theta}. \tag{19}$$

It can be seen from Eqs. (16b) and (18) that, at the interface, a smeared zero gradient ($\frac{\partial p}{\partial n} = 0$) boundary condition is applied on the pressure field. The boundary condition is smeared in the sense that the gradient gradually decreases across the interface and becomes zero at the end of the fluid domain. Finally, the Laplacian operator is defined only in Ω^{f}. The rows and columns in the discrete Laplacian, corresponding to the nodes in $\Omega^{s}\backslash\partial\Omega^{s}$, would be filled with zeros. Hence, while solving Eqs. (16b) and (18) those rows and columns have to be set inactive. This can be done in two ways. The efficient method is to discard these rows and columns from the solution procedure and solve a subsystem instead. In that case, the condition number of the matrix remains unchanged and the size of the linear system to be solved is reduced. The second option is to add unity to the diagonal entries of those nodes and solve the full system. However, depending on the number of nodes inside the solid domain the condition number of the matrix will be worsened.

In this paper, Eqs. (11), (13), (16), and (18) are solved using the fluid dynamics model Fluidity, an open-source finite element numerical tool that solves the Navier–Stokes equations on unstructured grids [30–32].

3.3. Treatment of moving interfaces

In a deforming/moving body problem, elements and nodes move across the fluid-structure interface. It is known that, for direct forcing immersed boundary methods, this can cause numerical oscillations [33]. One source of these oscillations comes from the difference in the order of the error between the fluid and solid domains. In the present method, both the solid and fluid domains are spatially discretised using the same shape functions. Hence, the numerical errors associated with both domains are of the same order of magnitude. However, the temporal discontinuity in the projection step (Eq. (18)) is amplified by a factor of $1/\Delta t$. This is illustrated by Fig. 4. The continuous circle highlights the position of the body at time level $n+1$, while the dashed circle outlines its position at time level n. At this time level, the node that is marked with a dark square (■) is internal to the solid domain. Pressure has no defined value inside the solid and is set to zero. However, when the body moves, this node becomes part of the fluid-solid interface and starts interacting with the fluid equations via the pressure gradient term. Since velocity has defined values inside the solid, operators that act on the velocity field are unaffected. However, errors are introduced in the evaluation of the pressure gradient and will propagate onto the velocity field. To remedy this, the pressure gradient term has to be moved entirely to the projection step. The modified iteration scheme is then

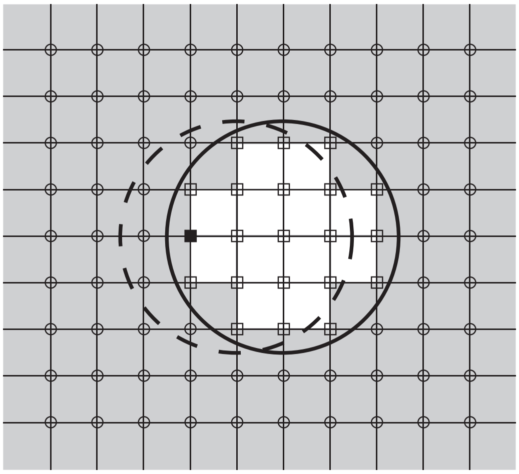


Fig. 4. Decomposed solid and fluid domains of a moving body on a finite element grid at time level $n+1$. Circular markers (○) represent fluid nodes ($\alpha^{n+1} = 0$). Square markers (□) represent solid nodes ($\alpha^{n+1} = 1$), where strong velocity boundary conditions are imposed. Shaded elements represent the domain in which regular fluid operators act.

$$(\mathbf{M} + \theta\Delta t(\mathbf{A} - \mathbf{D}))\frac{\mathbf{u}^{*} - \mathbf{u}^{n}}{\Delta t} = -(\mathbf{A} - \mathbf{D})\mathbf{u}^{n} + \mathbf{f} + \mathbf{f}^{s}, \tag{20}$$

$$\mathbf{L}p^{n+\theta} = \frac{-\bar{\mathbf{C}}^{T}(\theta\mathbf{u}^{*} + (1-\theta)\mathbf{u}^{n})}{\Delta t}, \tag{21}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{*} + \Delta t\mathbf{M}^{-1}\mathbf{C}p^{n+\theta}. \tag{22}$$

In that case, the Poisson's system solves the full pressure field (p) and not a correction for pressure (Δp).

4. Results

The method described in the previous section is applied to flows past fixed and moving structures. All the cases considered in this section are solved on two-dimensional unstructured grids with triangular elements and a CFL number equal to 0.95. The Crank-Nicolson method is used for temporal integration.

4.1. Taylor-Couette flow

Taylor-Couette flow is the flow between two concentric hollow cylinders [34]. The Reynolds number for this flow is defined based

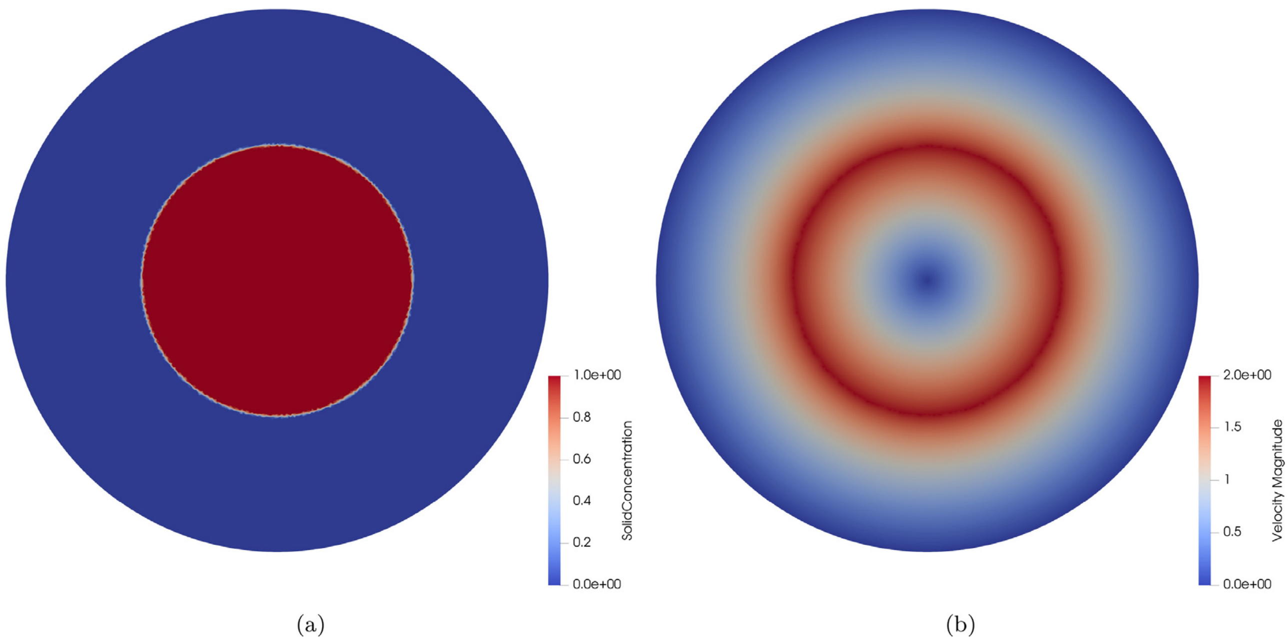


Fig. 5. Taylor-Couette flow results. (a) Projected solid concentration field on the fluid mesh, the domain shaded in red being the inner solid cylinder whilst the region of zero solid concentration (shaded in blue) being filled with fluid. (b) Computed velocity on the fluid mesh, where inside the solid domain, the velocity is equal to the projected solid velocity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

on the relative angular velocity of the cylinders and the diameter of one of the cylinders. At low Reynolds numbers, the flow is steady and purely azimuthal. The critical value of the Reynolds number corresponds to the point at which the flow becomes unstable and three-dimensional. Below the critical point, an analytical solution exists and is given by, in cylindrical coordinates,

$$u_\theta = Ar + \frac{B}{r}, \quad u_r = u_z = 0,$$
$$A = \Omega_{in}\frac{\mu - \eta^2}{1-\eta^2}, \quad B = \Omega_{in}R_{in}^2\frac{1-\mu}{1-\eta^2}$$
$$\mu = \frac{\Omega_{out}}{\Omega_{in}}, \quad \eta = \frac{R_{out}}{R_{in}}, \tag{23}$$

where R is the radius, Ω is the angular velocity, and subscripts *in* and *out* denote the cylinder to which the parameter belongs. The existence of an analytical solution makes Taylor-Couette flow an ideal case for grid convergence study. For the current test case, flow parameters are chosen as follows: $\Omega_{in} = 4, \Omega_{out} = 0, R_{in} = 0.5, R_{out} = 1,$ and the kinematic viscosity $\nu = 1$. The Reynolds number based on the diameter of the inner cylinder is $Re_{in} = 2,$ which is well below the known critical Reynolds number for this setup, i.e. $Re_{in}^{critical} = 68.4$.

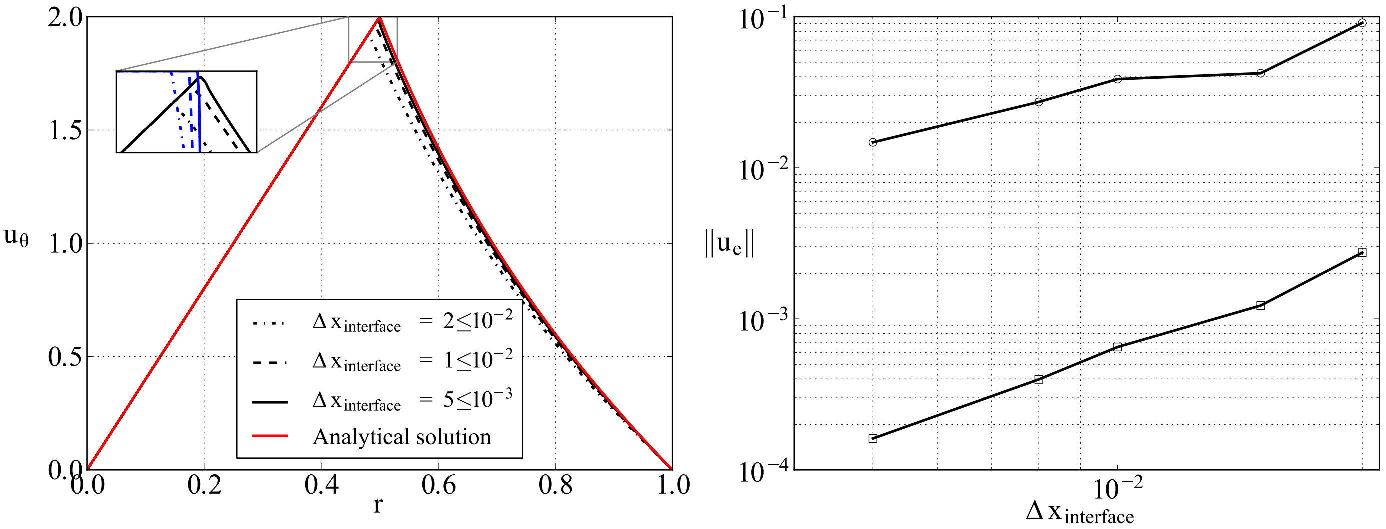
To model this problem using the current immersed boundary method, the hollow cylinders are replaced by solid discs. The motion of the inner cylinder is prescribed using an immersed solid, which rotates at an angular velocity of $\Omega_{in} = 4$. The domain shaded in red in Fig. 5a represents the solid inner disc that is projected onto the fluid mesh. Inside this solid domain, velocity is prescribed and varies linearly along the radial direction, as shown by Fig. 5b. The fixed outer cylinder is modelled using a regular body-conforming wall. The fluid field is initialised as at rest and the immersed body is impulsively set into motion.

Results are plotted at steady-state. Fig. 6a and b show the variation in tangential velocity in the radial direction. In Fig. 6a, the red line shows the exact solution and the black lines show the computed profiles at varying spatial resolutions. During the grid refinement study, the element edge length is varied only at the interface located at $r = 0.5,$ that is at the boundary of the inner cylinder. By contrast, near the outer-wall, the edge length is kept at a constant value of $\Delta x = 2 \times 10^{-2}$. The profile shows good agreement with the analytical solution and convergence is observed when the mesh density is increased. By focusing the increment in spatial resolution at the interface, it can be inferred from Fig. 6a that the error is a maximum at the interface ($r = 0.5$). This is a consequence of the conservative projection onto a non-matching grid. In the inset of Fig. 6a, it can be seen that the point from which the solid concentration field (blue lines) starts to drop varies depending on the mesh resolution. Because of the conservative projection, the radius of the projected cylinder is slightly smaller than the actual solid body. Since, in this particular case, the velocity boundary condition that has to be imposed at the fluid-structure interface is a function of the radius of the cylinder ($u_\theta(r)$), the imposed angular velocity at the interface is reduced. As shown in the figure, this effect disappears for sufficiently fine grids at the fluid-structure interface.

Fig. 6b shows the L_∞ (○) and L_2 (□) spatial velocity error ($\|u_e\|$) norms plotted against the element edge length at the fluid-structure interface. The spatial error norms are evaluated at steady-state by comparing the results with the analytical solution. Convergence study shows an order of about 1.31 and 2.06 for the L_∞ and L_2 error norms, respectively.

4.2. Flow past a stationary cylinder

In this section, flow past a stationary circular cylinder is considered as a test problem. As shown in Fig. 7, the computational domain has the dimensions 25d × 20d (d being the cylinder diameter) and the cylinder is centred at a distance 5d from the inlet and 10d from the sides. On the left boundary, that is marked with arrows pointing inwards, a uniform velocity profile is prescribed as inlet boundary condition. Open boundary conditions are applied on all the other domain boundaries. Fig. 7 also shows the characteristic dimensions of the wake structure as defined in Coutanceau and Bouard [36]. Dimensions l, a, and b represent the length of the recirculation zone, distance to the vortex core, and gap between the vortex cores, respectively. The separation angle



(a) (b)

Fig. 6. (a) Variation of tangential velocity in the radial direction in a Taylor-Couette flow. Red line shows the analytical solution [34]. Black lines show results using the current method at different grid resolutions close to the fluid-structure interface. Blue lines in the inset show the variation of the solid concentration field (α). (b) The L_∞: ○ and L_2: □ spatial velocity error ($\|u_e\|$) norms plotted against the element edge length at the interface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

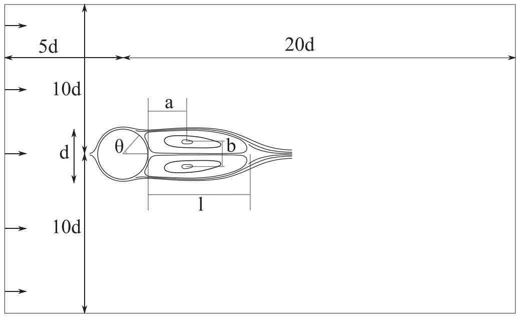


Fig. 7. Dimensions of the computational domain in terms of the cylinder diameter d and characteristic dimensions of the wake structure, as defined in [24].

is denoted by θ. Two Reynolds numbers are considered: $Re = 20$ and $Re = 40$. The steady-state results from the present immersed boundary method are compared with several results from the lit-

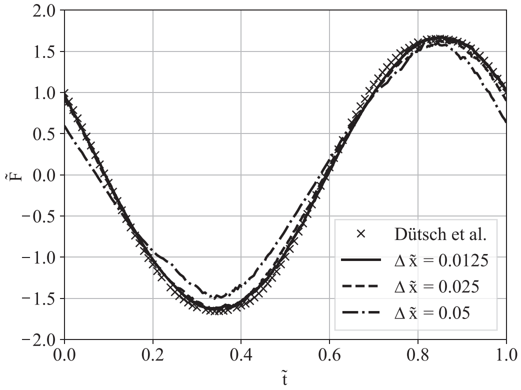


Fig. 9. Non-dimensionalised inline force acting on the cylinder in one cycle at Re=100 and KC=5. Black lines are results obtained using the immersed boundary method presented in this paper at varying grid resolutions near the interface. Black cross markers are from experimental data [40].

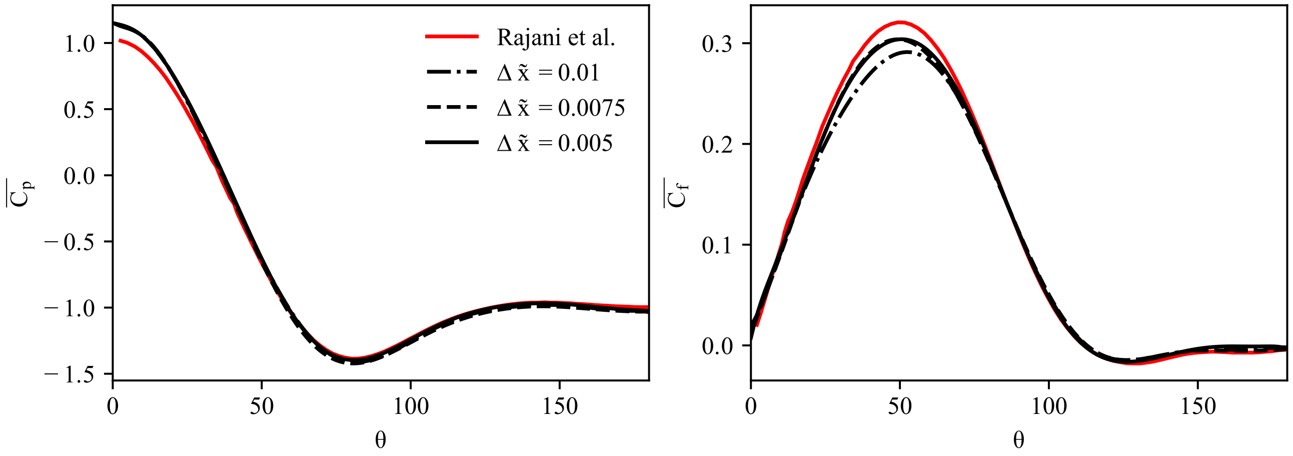
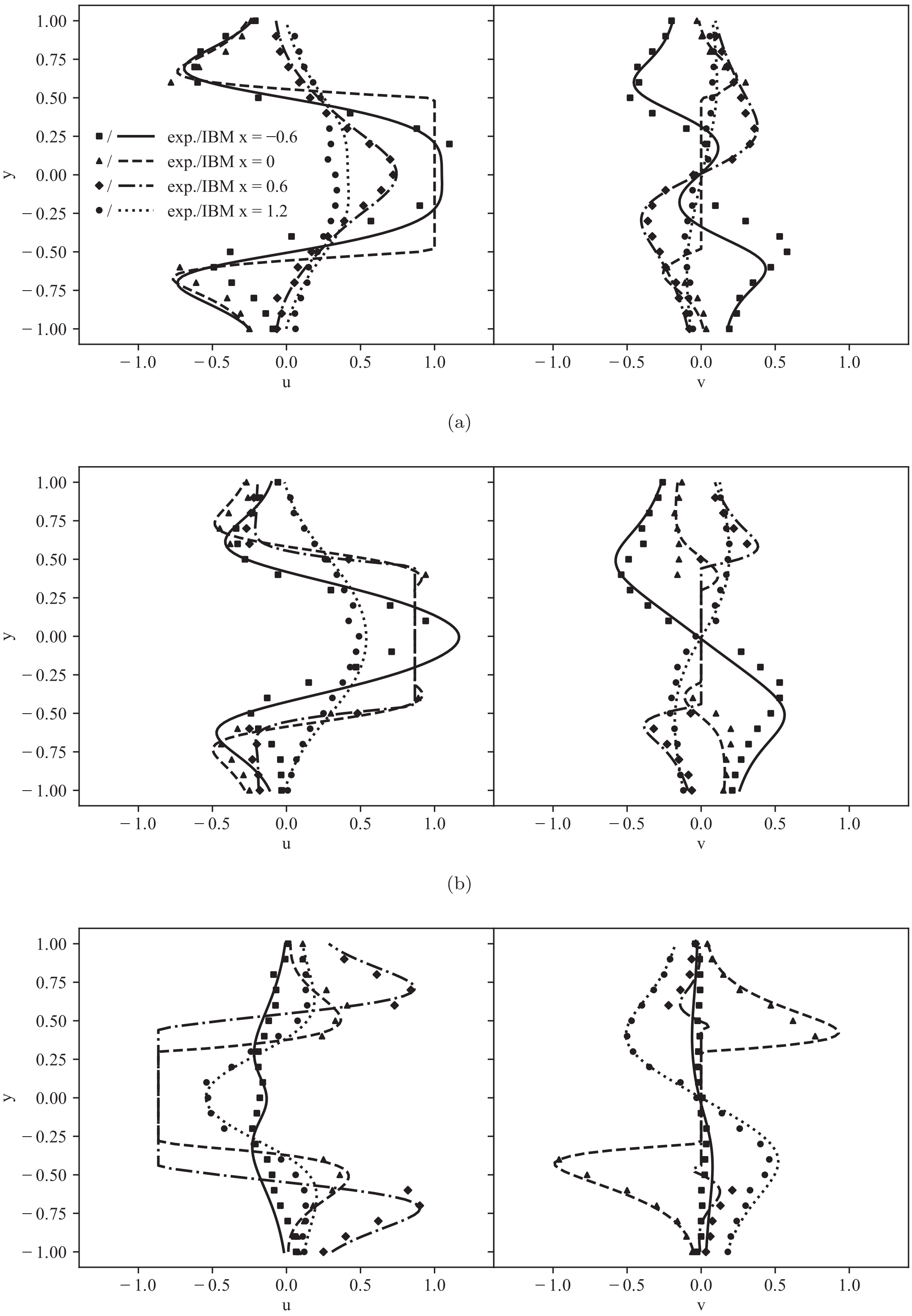


Fig. 8. Time averaged coefficients of pressure and friction from flow past a cylinder at a Reynolds number of 200 with varying edge lengths near the interface and compared to DNS data from existing literature [39]. θ is measured clockwise from the stagnation point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)

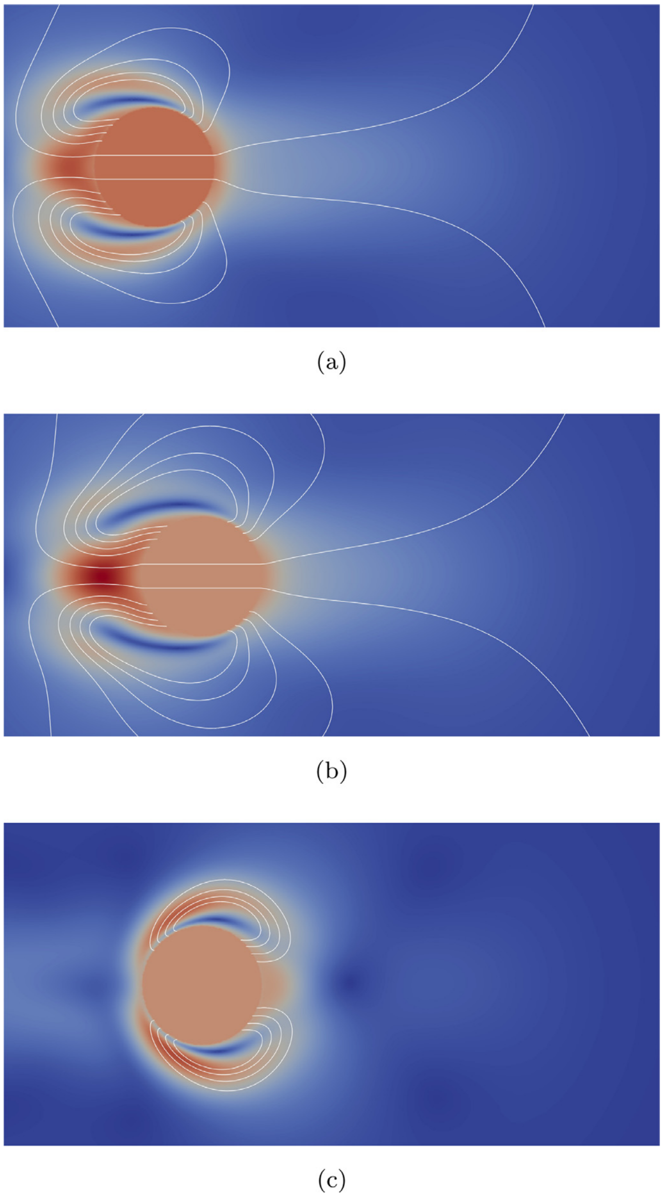
(b)

(c)

Fig. 10. Comparison of non-dimensionalised velocity components at four cross-sections between present computation (lines) and experimental data [40] (markers) at three different phases within one cycle of oscillation: (a) 180°, (b) 210°· and (c) 330°.

erature [24,35–38]. In particular, the characteristic lengths of the flow (non-dimensionalised by the cylinder diameter d) and the drag coefficient are presented in Table 1. It is shown that the flow profiles and the force coefficient from the present method agree well with those reported in the literature.

Furthermore, an unsteady test case is considered at a Reynolds number of 200. Fig. 8 shows time averaged pressure and friction coefficients along the periphery of the cylinder. Coefficients are averaged using a simple mean and θ is measured from the stagnation point. Results are plotted at three different grid edge lengths near the boundary of the cylinder: $l_e = 0.01d$ (dash-dot line), $l_e = 0.0075d$ (dash-dash line), and $l_e = 0.005d$ (solid black line). Results show grid covergence and agrees very well with data available from a body-conforming DNS simulation (red line) [39]. As with the steady cases, it can be noticed from the coefficient of friction curves that the average point of separation is predicted accurately.


(a)

(b)

(c)

Fig. 11. Velocity field and streamlines at three different phases within one cycle of oscillation: (a) 180°, (b) 210°, and (c) 330°.

Table 1
Steady state non-dimensionalised wake dimensions and coefficient of drag from flow past a cylinder at Reynolds numbers of 20 and 40.

		l/d	a/d	b/d	θ	C_D
$Re = 20$	Dennis and Chang [35]	0.94	–	–	43.7°	2.05
	Coutanceau and Bouard [36]	0.93	0.33	0.46	45.0°	–
	Tritton [37]	–	–	–	–	2.09
	Linnick and Fasel [38]	0.93	0.36	0.43	43.5°	2.06
	Taira and Colonius [24]	0.94	0.37	0.43	43.3°	2.06
	Present study	0.94	0.34	0.43	43.8°	2.05
$Re = 40$	Dennis and Chang [35]	2.35	–	–	53.8°	1.52
	Coutanceau and Bouard [36]	2.13	0.76	0.59	53.8°	–
	Tritton [37]	–	–	–	–	1.59
	Linnick and Fasel [38]	2.28	0.72	0.60	53.6°	1.54
	Taira and Colonius [24]	2.30	0.75	0.60	53.7°	1.54
	Present study	2.30	0.71	0.61	53.3°	1.55

4.3. Flow past an oscillating cylinder

In this section, the vortices generated by an oscillating cylinder placed in still fluid are simulated and compared against experimental data [40]. The cylinder is rigid and irrotational. The horizontal oscillation of the cylinder is defined by,

$$x(t) = -A\sin(2\pi ft), \quad u(t) = -2\pi fA\cos(2\pi ft), \tag{24}$$

where, $x(t)$ is the position of the centre of the cylinder and $u(t)$ is the velocity of the cylinder. The parameters A and f are the amplitude and frequency of the prescribed oscillation. Reynolds number and Keulegan-Carpenter number (KC) are further defined as,

$$Re = \frac{u_{max}d}{\nu}, \quad KC = \frac{u_{max}}{fd}. \tag{25}$$

For the simulation, the cylinder with diameter d is placed at the centre of a computational domain of size $55d \times 35d$. Reynolds number is set at 100 and Keulegan-Carpenter number is set at 5. Time (t) and horizontal force (F) are non-dimensionalised as $\tilde{t} = tf$ and $\tilde{F} = F/\rho d u_{\text{max}}$, respectively. Fig. 9 shows the variation of non-dimensionalised inline force ($\tilde{F}$) against non-dimensionalised time ($\tilde{t}$), for one period of oscillation, and at varying grid resolutions near the interface: $l_e = 0.05d$ (dash-dot line), $l_e = 0.025d$ (dash-dash line), and $l_e = 0.0125d$ (solid black line). The results have good agreement with the reference data [40] and shows grid convergence.

Fig. 10 compares velocity components of the flow field at four cross-sections along the line of oscillation with experimental data [40]. For the most part, the profiles agree very well. Where it dif-

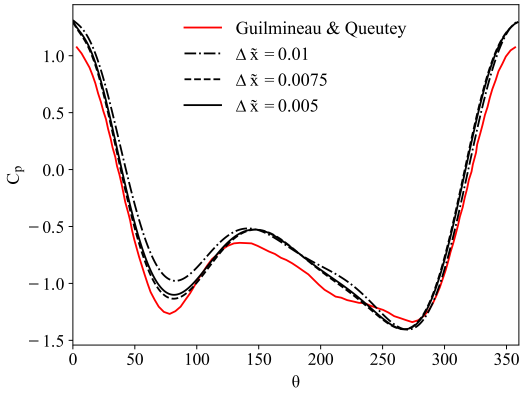


Fig. 12. Coefficient of pressure on the surface of the cylinder at a phase angle of 270° with varying edge lengths near the interface and compared to DNS data from existing literature [42]. θ is measured clockwise from the stagnation point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

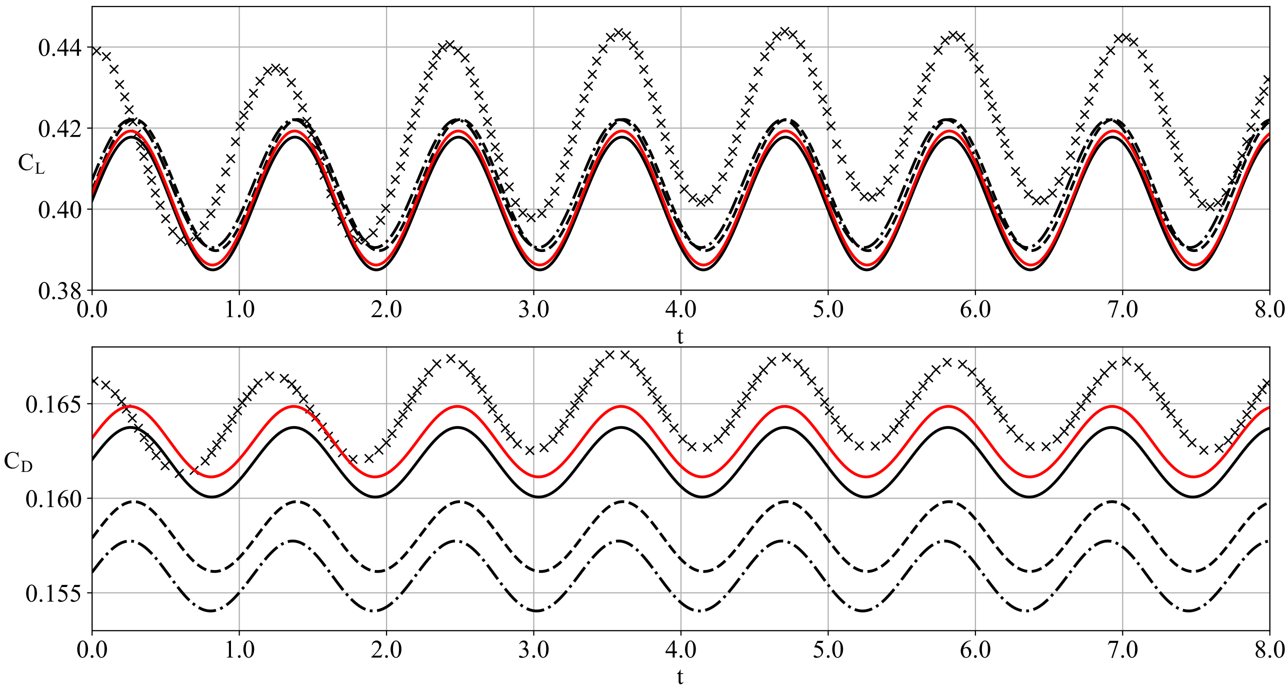


Fig. 13. Time evolution of the coefficients of lift (top) and drag (bottom) on a NACA0012 aerofoil at Re = 1000 and an angle of attack 10°. Black lines are results obtained using the immersed boundary method presented in this paper at varying grid resolutions near the interface. Red line is a body-conforming simulation result using the solver Fluidity, and black cross markers are DNS results from existing literature [43]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

fers, the mismatch is only on one side of the symmetric plane ($y = 0$) and is due to the asymmetry in the experimental data. In Fig. 11, streamlines in the vicinity of the cylinder and the velocity field are plotted at different phases in the cycle. Streamline curves are a good match, and can be compared against plots from the reference [40] at the same phase in the cycle.

As an extension, we now add a high enough background velocity to the previously quiescent fluid such that the cylinder can naturally shed vortices even when it is stationary. From here on, we switch back to the flow velocity based definition for Reynolds number (instead of $u_{\max}$ of the oscillation), and use Strouhal number (instead of Keulegan-Carpenter number) to define our test case. For our test case, we set the Reynolds number at 185, and this gives us a Strouhal number of 0.193 [41]. Additionally, the amplitude of oscillation (A) is set as $0.2d$, and the ratio of the frequencies of excitation and natural vortex shedding (f/f_n) is set to 0.9. Fig. 12 plots the coefficient of pressure on the surface of the cylinder at a phase angle of 270°, (centre of the cylinder, $x(t) = A$) for three different grid resolutions near the immersed interface: $l_e = 0.01d$ (dash-dot line), $l_e = 0.0075d$ (dash-dash line), and $l_e = 0.005d$ (solid black line). The results match with the reference data (red line) [42] and shows grid convergence.

4.4. Flow past a NACA0012 aerofoil

Flow past the NACA0012 aerofoil is considered to demonstrate the capabilities of the method with moderately turbulent flows. The chord-based Reynolds number is 1000. The size of the computational domain is the same as in Fig. 7, except that dimensions are here expressed in terms of the chord length (c). The boundary conditions are also the same. Fig. 13 shows the time evolution of the lift and drag coefficients at an angle of attack 10° using the present immersed boundary method at three grid edge lengths near the aerofoil boundary: $l_e = 0.0025c$ (dash-dot line), $l_e = 0.00175c$ (dash-dash line), and $l_e = 0.001c$ (solid black line). The results are further compared with a body-conforming DNS available in the literature [43] (black cross markers) and a body-conforming simulation using Fluidity and $l_e = 0.0025c$ (red line). It can be seen from the plots that there is a difference in amplitude of the oscillations between the body conforming simulation from Fluidity (red line) and the DNS data obtained from literature (black cross markers). The relative difference in magnitude between the body-conforming results is maximum 4% for the drag coefficient and 1% for the lift coefficient. This could be explained by the use of linear shape functions for both velocity and pressure fields in the present simulations and the use of fully unstructured meshes. The agreement between the present immersed boundary method and the body-conforming simulation using Fluidity is however very good. In particular, the lift coefficients from all the Fluidity simulations almost match perfectly with one another. The drag coefficient is more sensitive to the mesh resolution used near the aerofoil. The drag coefficient given by the immersed boundary method deviates by ~ 5% with the coarser mesh (dash-dot line) compared to the body-conforming case (red line), and this percentage decreases as the mesh is refined, to ~ 3% for the intermediate mesh and ~ 1% for the finest mesh.

5. Conclusions

This paper presents a new formulation of the immersed boundary method that is specific to finite-element discretisation methods. The proposed method modifies the test functions in such a way that the spaces occupied by the fluid and solid, respectively, are decomposed into two sub-domains. Unlike many of the forcing immersed boundary methods available in the literature, the present method satisfies both the divergence-free and no-slip conditions without introducing an additional forcing term in the momentum equations of the fluid. The absence of a forcing term implies that the overall CFL limit is not affected by the immersed boundary implementation. Furthermore, the modified test functions preserve symmetry and positive definiteness of the Laplacian operator, so that the Poisson's system can be efficiently solved using existing numerical methods.

It is shown that the method provides accurate results on a series of test cases with both fixed and moving bodies, as well as laminar and moderately turbulent flows. In particular, excellent agreement was found between the present method and experimental data for low-Reynolds numbers flows. For flow past an aerofoil at a higher Reynolds number, an accurate prediction of the lift coefficient was obtained at the same grid size requirement of a body-conforming simulation. The present method however required a slightly finer mesh resolution compared to a body-conforming method, in order to provide a drag coefficient of equal accuracy. Future work will focus on extending the present method to highly turbulent flows of interest for engineering applications.

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.

CRediT authorship contribution statement

Navaneetha Krishnan: Conceptualization, Methodology, Writing - original draft. **Axelle Viré:** Conceptualization, Writing - original draft, Supervision, Funding acquisition. **Roland Schmehl:** Supervision, Writing - review & editing, Funding acquisition. **Gerard van Bussel:** Resources, Supervision.

Acknowledgements

A. Viré and N. Krishnan are financially supported by the European Union Seventh Framework Programme (FP7/2007-2013) under a Marie Curie Career Integration Grant (Grant Agreement PCIG13-GA-2013-618159). A. Viré also acknowledges support from the Rijksdienst voor Ondernemend Nederland (RVO) through the TSE Hernieuwbare Energie funding scheme (ABIBA project - TEHE116332). R. Schmehl is financially supported by the European Commission Horizon 2020 framework programme under AWESCO (H2020-ITN-642682) and REACH (H2020-FTIPilot-691173) projects. This work was partly carried out on the Dutch national e-infrastructure with the support of SURF Cooperative (project number 17215).

References

[1] Iaccarino G, Verzicco R. Immersed boundary technique for turbulent flow simulations. Appl Mech Rev 2003;56(3):331–47. doi:10.1115/1.1563627.

[2] Mittal R, Iaccarino G. Immersed boundary methods. Annu Rev Fluid Mech 2005;37(1):239–61. doi:10.1146/annurev.fluid.37.061903.175743.

[3] Sotiropoulos F, Yang X. Immersed boundary methods for simulating fluid-structure interaction. Prog Aerosp Sci 2014;65:1–21. doi:10.1016/j.paerosci.2013.09.003.

[4] Peskin CS. Flow patterns around heart valves: a numerical method. J Comput Phys 1972;10(2):252–71. doi:10.1016/0021-9991(72)90065-4.

[5] Arquis E, Caltagirone JP. Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieux poreux: application à la convection naturelle. In: Comptes rendus de l'academie des sciences, 299; 1984. p. 1–4.

[6] Kolomenskiy D, Schneider K. A fourier spectral method for the Navier-Stokes equations with volume penalization for moving solid obstacles. J Comput Phys 2009;228(16):5687–709. doi:10.1016/j.jcp.2009.04.026.

[7] Gazzola M, Chatelain P, van Rees WM, Koumoutsakos P. Simulations of single and multiple swimmers with non-divergence free deforming geometries. J Comput Phys 2011;230(19):7093–114. doi:10.1016/j.jcp.2011.04.025.

[8] Boffi D, Gastaldi L. A finite element approach for the immersed boundary method. Comput Struct 2003;81(8):491–501. doi:10.1016/S0045-7949(02)00404-2.

[9] Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. Comput Methods Appl Mech Eng 2004;193(21):2051–67. doi:10.1016/j.cma.2003.12.044.

[10] Liu WK, Liu Y, Farrell D, Zhang L, Wang XS, Fukui Y, et al. Immersed finite element method and its applications to biological systems. Comput Methods Appl Mech Eng 2006;195(13):1722–49. doi:10.1016/j.cma.2005.05.049.

[11] Viré A, Xiang J, Pain CC. An immersed-shell method for modelling fluid-structure interactions. Philos Trans R Soc Lond A 2015;373(2035):1–14. doi:10.1098/rsta.2014.0085.

[12] Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force field. J Comput Phys 1993;105(2):354–66. doi:10.1006/jcph.1993.1081.

[13] Mohd-Yusof J. Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries. In: Center for turbulence research annual research briefs. Stanford University; 1997. p. 317–27.

[14] Fadlun E, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J Comput Phys 2000;161(1):35–60. doi:10.1006/jcph.2000.6484.

[15] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. J Comput Phys 2005;209(2):448–76. doi:10.1016/j.jcp.2005.03.017.

[16] Su S-W, Lai M-C, Lin C-A. An immersed boundary technique for simulating complex flows with rigid boundary. Comput Fluids 2007;36(2):313–24. doi:10.1016/j.compfluid.2005.09.004.

[17] Ye T, Mittal R, Udaykumar H, Shyy W. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. J Comput Phys 1999;156(2):209–40. doi:10.1006/jcph.1999.6356.

[18] Kirkpatrick M, Armfield S, Kent J. A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid. J Comput Phys 2003;184(1):1–36. doi:10.1016/s0021-9991(02)00013-x.

[19] Cheny Y, Botella O. The ls-stag method: a new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. J Comput Phys 2010;229(4):1043–76. doi:10.1016/j.jcp.2009.10.007.

[20] Meyer M, Devesa A, Hickel S, Hu X, Adams N. A conservative immersed interface method for large-eddy simulation of incompressible flows. J Comput Phys 2010;229(18):6300–17. doi:10.1016/j.jcp.2010.04.040.

[21] Kim J, Kim D, Choi H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. J Comput Phys 2001;171(1):132–50. doi:10.1006/jcph.2001.6778.

[22] Ge L, Sotiropoulos F. A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries. J Comput Phys 2007;225(2):1782–809. doi:10.1016/j.jcp.2007.02.017.

[23] Glowinski R, Pan T, Priaux J. Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. Comput Methods Appl Mech Eng 1998;151(1):181–94. doi:10.1016/s0045-7825(97)00116-3.

[24] Taira K, Colonius T. The immersed boundary method: a projection approach. J Comput Phys 2007;225:2118–37. doi:10.1016/j.jcp.2007.03.005.

[25] Boffi D, Cavallini N, Gastaldi L. The finite element immersed boundary method with distributed lagrange multiplier. SIAM J Numer Anal 2015;53(6):2584–604. doi:10.1137/140978399.

[26] Elman HC, Silvester DJ, Wathen AJ. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Numerical Mathematics and Scientific Computation. Oxford University Press; 2005. doi:10.1093/acprof:oso/9780199678792.001.0001.

[27] Farrell PE, Piggott MD, Pain CC, Gorman G, Wilson C. Conservative interpolation between unstructured meshes via supermesh construction. Comput Methods Appl Mech Eng 2009;198(33–36):2632–42. doi:10.1016/j.cma.2009.03.004.

[28] Viré A, Xiang J, Milthaler F, Farrell PE, Piggott MD, Latham J-P, et al. Modelling of fluid-solid interactions using an adaptive mesh fluid model coupled with a combined finite discrete element model. Ocean Dyn 2012;62(10–12):1487–501. doi:10.1007/s10236-012-0575-z.

[29] Chorin AJ. Numerical solution of the Navier-Stokes equations. Math Comput 1968;22:745–62. doi:10.1090/S0025-5718-1968-0242392-2.

[30] Pain CC, Umpleby AP, de Oliveira C, Goddard AJH. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. Comput Methods Appl Mech Eng 2001;190:3771–96. doi:10.1016/S0045-7825(00)00294-2.

[31] Pain C, Piggott M, Goddard A, Fang F, Gorman G, Marshall D, et al. Three-dimensional unstructured mesh ocean modelling. Ocean Modell 2005;10(1–2):5–33. doi:10.1016/j.ocemod.2004.07.005.

[32] Piggott MD, Gorman GJ, Pain CC, Allison PA, Candy AS, Martin BT, et al. A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes. Int J Numer Methods Fluids 2008;56:1003–15. doi:10.1002/fld.1663.

[33] Luo H, Dai H, de Sousa PJF, Yin B. On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries. Comput Fluids 2012;56:61–76. doi:10.1016/j.compfluid.2011.11.015.

[34] Taylor GI. Stability of a viscous liquid contained between two rotating cylinders. Philos Trans R Soc A 1923;223(605–615):289–343. doi:10.1098/rsta.1923.0008.

[35] Dennis SCR, Chang G-Z. Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. J Fluid Mech 1970;42(3):471–89. doi:10.1017/S0022112070001428.

[36] Coutanceau M, Bouard R. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. J Fluid Mech 1977;79(2):231–56. doi:10.1017/S0022112077000135.

[37] Tritton DJ. Experiments on the flow past a circular cylinder at low Reynolds numbers. J Fluid Mech 1959;6(4):547–67. doi:10.1017/S0022112059000829.

[38] Linnick M, Fasel H. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. J Comput Phys 2005;204(1):157–92. doi:10.1016/j.jcp.2004.09.017.

[39] Rajani B, Kandasamy A, Majumdar S. Numerical simulation of laminar flow past a circular cylinder. Appl Math Model 2009;33(3):1228–47. doi:10.1016/j.apm.2008.01.017.

[40] Dütsch H, Durst F, Becker S, Lienhart H. Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers. J Fluid Mech 1998;360:249–71. doi:10.1017/s002211209800860x.

[41] Williamson CHK. Defining a universal and continuous Strouhal-Reynolds number relationship for the laminar vortex shedding of a circular cylinder. Phys Fluids 1988;31(10):2742–4. doi:10.1063/1.866978.

[42] Guilmineau E, Queutey P. A numerical simulation of vortex shedding from an oscillating circular cylinder. J Fluids Struct 2002;16(6):773–94. doi:10.1006/jfls.2002.0449.

[43] Mittal S, Tezduyar TE. Massively parallel finite element computation of incompressible flows involving fluid-body interactions. Comput Methods Appl Mech Eng 1994;112(1):253–82. doi:10.1016/0045-7825(94)90029-9.