

An Immersed Interface Method for Discrete Surfaces

Ebrahim M. Kolahdouz^{1,2}, Amneet Pal Singh Bhalla³, Brent A. Craven², and Boyce E. Griffith^{4,5,6}

¹Department of Mathematics, University of North Carolina, Chapel Hill, NC, USA

²Division of Applied Mechanics, Office of Science and Engineering Laboratories, Center for Devices and Radiological Health, United States Food and Drug Administration, Silver Spring, MD, USA

³Department of Mechanical Engineering, San Diego State University, San Diego, CA, USA

⁴Departments of Mathematics, Applied Physical Sciences, and Biomedical Engineering, University of North Carolina, Chapel Hill, NC, USA

⁵Carolina Center for Interdisciplinary Applied Mathematics, University of North Carolina, Chapel Hill, NC, USA

⁶McAllister Heart Institute, University of North Carolina, Chapel Hill, NC, USA
 ebrahimk@email.unc.edu and boyceg@email.unc.edu

June 5, 2019

Abstract

Fluid-structure systems occur in a range of scientific and engineering applications. The immersed boundary (IB) method is a widely recognized and effective modeling paradigm for simulating fluid-structure interaction (FSI) in such systems, but a difficulty of the IB formulation of these problems is that the pressure and viscous stress are generally discontinuous at fluid-solid interfaces. The conventional IB method regularizes these discontinuities, which typically yields low-order accuracy at these interfaces. The immersed interface method (IIM) is an IB-like approach to FSI that sharply imposes stress jump conditions, enabling higher-order accuracy, but prior applications of the IIM have been largely restricted to numerical methods that rely on smooth representations of the interface geometry. This paper introduces an immersed interface formulation that uses only a C^0 representation of the immersed interface, such as those provided by standard nodal Lagrangian finite element methods. Verification examples for models with prescribed interface motion demonstrate that the method sharply resolves stress discontinuities along immersed boundaries while avoiding the need for analytic information about the interface geometry. Our results also demonstrate that only the lowest-order jump conditions for the pressure and velocity gradient are required to realize global second-order accuracy. Specifically, we demonstrate second-order global convergence rates along with nearly second-order local convergence in the Eulerian velocity field, and between first- and second-order global convergence rates along with approximately first-order local convergence for the Eulerian pressure field. We also demonstrate approximately second-order local convergence in the interfacial displacement and velocity along with first-order local convergence in the fluid traction along the interface. As a demonstration of the method's ability to tackle more complex geometries, the present approach is also used to simulate flow in a patient-averaged anatomical model of the inferior vena cava, which is the large vein that carries deoxygenated blood from the lower extremities back to the heart. Comparisons of the general hemodynamics and wall shear stress obtained by the present IIM and a body-fitted discretization approach show that the present method yields results that are in good agreement with those obtained by the body-fitted approach.

Keywords: immersed boundary method, immersed interface method, finite element, fluid-structure interaction, jump conditions, complex geometries

1 Introduction and overview

Stable and accurate yet simple and computationally tractable methods for treating interfaces are of great importance in fluid dynamics problems involving immersed boundaries. Body-fitted discretization approaches, which approximate the fluid dynamics on a computational domain that conforms to the interface geometry, can provide high accuracy for such problems but are limited by the continued difficulties of mesh generation, especially for problems involving substantial motion of the immersed interface. A widely used alternative approach to these types of interactions is the immersed boundary (IB) method introduced by Peskin [1, 2]. Unlike body-fitted methods, the IB approach avoids dynamic mesh regeneration and allows for the use of fast structured-grid fluid solvers. In the classical IB method, the Navier-Stokes equations are solved on a fixed background Eulerian grid, and the immersed structure is represented as a collection of Lagrangian markers. The interaction of Eulerian and Lagrangian frameworks takes place in two steps: 1) spreading structural forces from the Lagrangian markers to the Eulerian grid using a regularized Dirac delta function and 2) interpolating velocities from the Eulerian grid back to the Lagrangian markers using the same smoothed delta function.

A limitation of the conventional IB method is that it yields low-order accuracy at fluid-structure interfaces. The continuous form of the IB equations uses integral equations with singular kernels to connect the Lagrangian and Eulerian frames, and this formulation is equivalent to a formulation involving jump conditions [3]. However, the use of regularized delta functions in the conventional numerical realization of the IB method has the effect of regularizing stress jumps at those interfaces, which implies that stresses do not converge pointwise at the interface. Much work to improve the IB method has focused on improving its accuracy while retaining the original method’s use of nonconforming grids. The first formally second-order version of the method was introduced by Lai and Peskin [4]. This method was formulated to be second-order accurate if applied to a hypothetical problem in which the regularized delta functions are replaced by fixed smooth functions, independent of the mesh [4]. This method was refined by Griffith and Peskin and applied to a specific FSI problem for which the formally second-order accurate method was able to achieve its designed order of accuracy [5], although for general FSI problems, the formally second-order accurate method still only realizes first-order convergence rates. This approach was further extended by Griffith et al. [6] to use adaptive mesh refinement. This methodology has enabled modeling in a number of application areas, including cardiac dynamics [7, 8, 9, 10, 11, 12, 13, 14, 15], platelet adhesion [16], esophageal transport [17, 18, 19], heart development [20, 21], insect flight [22, 23], and undulatory swimming [24, 25, 26, 27, 28, 29, 30].

Meanwhile, motivated by the same objective, a number of sharp interface methods have also been developed. Among them are the immersed interface method (IIM) [31, 32, 33], the ghost-fluid method [34, 35], the cut-cell method [36, 37, 38, 39, 40], the hybrid Cartesian-immersed boundary method [41, 42], and the curvilinear immersed boundary method [43, 44]. Many of these methods achieve higher-order accuracy by adopting approaches that are similar to body-fitted discretization methods, such as local modifications to the finite difference stencils, to allow for the accurate reconstruction of boundary conditions in the vicinity of the immersed interface.

This paper introduces a new immersed interface scheme for the incompressible Navier-Stokes equations that uses only a C^0 -continuous representation of the interface geometry. Immersed interface methods resemble the classical IB method in that the motion is derived from a singular surface force. In the classical IB method and other continuum surface force approaches [45, 46], these forces are smoothed out over a small region near the interface using a smooth kernel function. In the IIM, these forces are used to determine discrete jump conditions that are imposed in the finite difference discretization of the fluid equations. In fact, in most modern IIMs for FSI, generalized Taylor series expansions are used to extend the jump values from the interface to the finite difference discretization [47]. In this work, we view this procedure as constructing a discrete force-coupling operator that is tailored to a particular fluid discretization method.

The IIM was introduced by Leveque and Li [31] for the solution of elliptic PDEs with discontinuous coefficients, or in the presence of singular forces. This initial method was extended to the two-dimensional solution of the incompressible Stokes [48, 49] and Navier-Stokes [32, 33, 50] equations, typically in combination with geometrical representations of the interface based on level set methods [51, 52, 53]. The systematic derivation of the jump conditions for the velocity and its first and second normal derivatives, as well as the pressure and its first normal derivative at the interface, was detailed by Lai and Li [3]. They then coupled their method with a second-order accurate projection algorithm to solve the full Navier-Stokes equations

in two spatial dimensions and empirically demonstrated that the selected jump conditions were adequate to achieve full second-order accuracy for the velocity and nearly second-order accuracy for the pressure in the maximum norms [32]. Later, Xu and Wang used a generalized surface parameterization for the interface representation and derived jump conditions for the first-, second-, and third-order spatial derivatives of velocity and pressure along with jump conditions in the first- and second-order time derivatives of the velocity [47]. They used their method in both two-dimensional [54] and three-dimensional [55] applications. In the infinity norm, second-order accuracy of both velocity and pressure was demonstrated in two spatial dimensions, although some deviations from the designed order of accuracy were observed in empirical tests. In the three-dimensional cases, nearly second-order accuracy was shown in the infinity norm of the velocity, and between first- and second-order accuracy was observed for the pressure. Local truncation error analysis suggests that achieving pointwise second-order accuracy in solving an elliptic PDE with a solution that includes a discontinuity along an internal interface requires correction terms up to the third normal derivative [56, 54]. However, several authors have reported empirical results demonstrating local second-order accuracy when including jumps only up to the second normal derivative, provided that all the spatial terms in the original equation are approximated to second-order accuracy [32, 50, 57]. Because the interfacial discontinuities are one dimension lower than the solution domain, prior work has also studied how these conditions are sufficient despite the truncation error being reduced to first-order at the interface [56, 57]. The textbook of Li and Ito [56] provides additional details on the IIM, which is now routinely used to simulate various physical phenomena [51, 58, 59, 60, 61, 62].

Almost all previous work using the IIM represents the immersed boundary as a smooth interface, and in many cases, it is assumed that an analytic description of the interface geometry is available. An exception is the work of Xu and Wang [55], in which triangular patches are used to find the intersections between the interface and the finite difference stencils. However, in that work, the boundary parameterization still relies on an analytic description of the interface for computing geometrical quantities [55], which limits the applicability of that method to situations with more general boundary geometries.

The present numerical method does not use analytical information about the interface geometry. Instead, all geometrical information is provided by the discrete finite element representation of the interface. The ability to use such representations facilitates simulations with complex interfacial geometries. However, because even the lowest-order jump conditions require geometrical quantities that are discontinuous on a C^0 representation (e.g. the normal vector to the surface), the use of these representations requires a substantial extension of prior IIM formulations. In the present work, an L^2 projection is used to construct jump condition values that are continuous along the C^0 interface. An L^2 projection is also used to construct a velocity interpolation scheme that accounts for the known velocity discontinuities along the interface.

Empirical results from verification tests demonstrate that the method attains second-order global convergence rates along with nearly second-order local convergence in the Eulerian velocity field, and between first- and second-order global convergence rates along with approximately first-order local convergence for the Eulerian pressure field. These tests also show that along the interface, the method yields approximately second-order local accuracy in the velocity along with first-order local accuracy in the fluid traction (pressure and wall shear stress). In this work, we impose only the lowest order jump conditions in the pressure and velocity gradient. To our knowledge, all prior work using the IIM that has achieved similar levels of accuracy has imposed additional higher-order jump conditions. We also show that this approach yields a substantial improvement in accuracy as compared to a conventional formally second-order accurate IB method, and that it is necessary to use both pressure and velocity jump conditions along with corrected velocity interpolation for the present immersed interface scheme to achieve its full accuracy. As a demonstration of the method's versatility in treating complex geometries, this work also presents initial results of IIM simulations of flow in a patient-averaged anatomical model of the human inferior vena cava, which is the major vein that is responsible for returning deoxygenated blood from the lower extremities back to the heart.

2 Mathematical formulation

This section introduces the governing equations and physical interface conditions in the continuous formulation of the IB method. The alternative treatment of singular forces in the form of jump conditions for the pressure and velocity derivatives is then described. The resulting jump conditions will be used in the

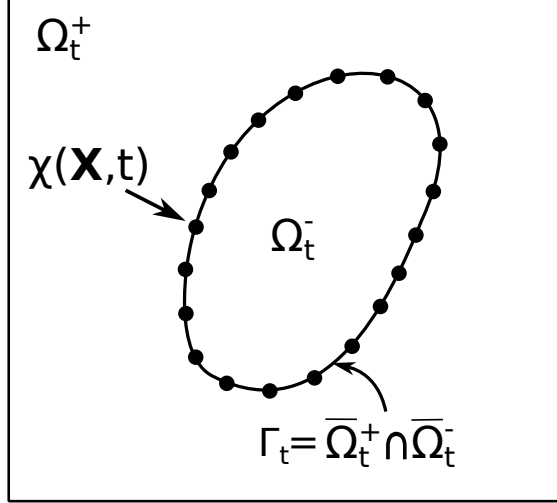


Figure 1: Lagrangian and Eulerian coordinate systems for a thin massless interface undergoing infinitesimal displacements. At time t , the physical position of a material point $\mathbf{X} \in \Gamma_0$ attached to the interface and parameterized by $\mathbf{X} = \mathbf{X}(\iota, \nu)$ is $\chi(\mathbf{X}, t) \in \Gamma_t$, in which (ι, ν) are curvilinear coordinates. In practice, the C^0 interface discretization is the only geometrical information used in the numerical method.

discretization technique described in Sec. 3. Throughout the paper, Eulerian quantities are indicated using lower case variables, and Lagrangian quantities are indicated by upper case variables. Bold face variables are used for vectors, and bold double-bar symbols are used for tensors. Superscripts ‘+’ (‘-’) indicate values obtained from the ‘exterior’ (‘interior’) side of an interface.

2.1 Governing equations

This study concerns the flow of a viscous, incompressible Newtonian fluid interacting with an infinitesimally thin immersed boundary. The computational domain is Ω , and $\mathbf{x} \in \Omega$ indicates fixed physical coordinates. We consider the case in which Ω is divided into an exterior region $\Omega^+ = \Omega_t^+$ and an interior region $\Omega^- = \Omega_t^-$, each parameterized by time t . We require that $\Omega = \Omega_t^+ \cup \Omega_t^-$, so that the interface is $\Gamma_t = \bar{\Omega}_t^+ \cap \bar{\Omega}_t^-$. Here, we consider only cases in which the interface is stationary or moves in a prescribed manner. We describe the motion of the interface through reference coordinates $\mathbf{X} \in \Gamma_0$ attached to the interface at time $t = 0$. To simplify the numerical treatment of this problem, we use a penalty method that allows for small deviations between the prescribed position of the interface and its actual physical configuration. The prescribed physical position of material point \mathbf{X} at time t is $\xi(\mathbf{X}, t)$, which has velocity $\mathbf{W} = \partial \xi(\mathbf{X}, t) / \partial t$, and the actual physical position of material point \mathbf{X} at time t is $\chi(\mathbf{X}, t)$, which has velocity $\mathbf{U}(\mathbf{X}, t) = \partial \chi(\mathbf{X}, t) / \partial t$. Because there will generally be a discrepancy between the prescribed and actual configurations of the boundary, we use the actual configuration, which is determined by the computed motion map $\chi : \Gamma_0 \times t \mapsto \Omega$, to determine the location where jump conditions are imposed and where the interface velocity is evaluated, see Fig. 1. The interface can also be described using curvilinear coordinates, say (ι, ν) , so that $\mathbf{X} = \mathbf{X}(\iota, \nu)$ and $\chi = \chi(\mathbf{X}(\iota, \nu), t)$. The present numerical method does not use a global curvilinear coordinate system. The finite element discretization of the interface geometry implicitly defines a local curvilinear coordinate system (i.e., in terms of the coordinates of the reference elements), but because this is a mesh-dependent coordinate system, we describe the interface geometry using reference coordinates attached to the interface. This approach leads to mesh-independent parameters and can be readily implemented within an isoparametric finite element formulation.

We consider the case in which both the mass density ρ and dynamic viscosity μ of the fluid are uniform across Ω . The effect of the immersed boundary appears as a singular force distribution in the equation of

momentum balance. In this case, the governing equations are

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, t) + \mathbf{u}(\mathbf{x}, t) \cdot \nabla \mathbf{u}(\mathbf{x}, t) \right) = -\nabla p(\mathbf{x}, t) + \mu \nabla^2 \mathbf{u}(\mathbf{x}, t) + \mathbf{f}(\mathbf{x}, t) \quad (1)$$

$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0, \quad (2)$$

$$\mathbf{f}(\mathbf{x}, t) = \int_{\Gamma_0} \mathbf{F}(\mathbf{X}, t) \delta(\mathbf{x} - \boldsymbol{\chi}(\mathbf{X}, t)) dA, \quad (3)$$

$$\frac{\partial \boldsymbol{\chi}}{\partial t}(\mathbf{X}, t) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \boldsymbol{\chi}(\mathbf{X}, t)) d\mathbf{x}, \quad (4)$$

in which $\mathbf{u}(\mathbf{x}, t)$ is the velocity, $p(\mathbf{x}, t)$ is the pressure, $\delta(\mathbf{x}) = \prod_{i=1}^d \delta(x_i)$ is the d -dimensional delta function, $\mathbf{F}(\mathbf{X}, t)$ is the interfacial force along the immersed boundary, and $\mathbf{f}(\mathbf{x}, t)$ is the corresponding singular Eulerian force density. Notice that interaction between the interface and the fluid occurs along $\Gamma_t = \boldsymbol{\chi}(\Gamma_0, t)$. Additionally in Eq. (4), $\mathbf{U}(\mathbf{X}, t) = \partial \boldsymbol{\chi}(\mathbf{X}, t) / \partial t = \mathbf{u}(\boldsymbol{\chi}(\mathbf{X}, t), t)$ because the velocity is continuous across the boundary as a result of the no-slip and no-penetration conditions along the interface.

For a rigid interface or an interface with prescribed kinematics, the interfacial force density is a Lagrange multiplier for the imposed motion. In this study, we consider a penalty formulation similar to that proposed by Goldstein et al. [63]. In this formulation, the rigidity constraint is inexactly imposed through an approximate Lagrange multiplier force,

$$\mathbf{F}(\mathbf{X}, t) = \kappa (\boldsymbol{\xi}(\mathbf{X}, t) - \boldsymbol{\chi}(\mathbf{X}, t)) + \eta (\mathbf{W}(\mathbf{X}, t) - \mathbf{U}(\mathbf{X}, t)). \quad (5)$$

The first term on the right-hand side is a linear spring force, with κ a spring stiffness penalty parameter, and the second term is a damping force, with η a damping penalty parameter. Both forces act to provide energetic penalization if the structure deviates from its prescribed position. Note that as $\kappa \rightarrow \infty$, the formulation imposes a hard constraint on the deformation. Numerical experiments using both the standard IB method and the immersed interface approach also demonstrate that $\eta > 0$ can help reduce numerical oscillations, particularly in problems with high pressure forces. In the special case of a stationary interface, $\boldsymbol{\xi}(\mathbf{X}, t) = \mathbf{X}$ and $\mathbf{W}(\mathbf{X}, t) \equiv \mathbf{0}$, and the penalty force becomes simply

$$\mathbf{F}(\mathbf{X}, t) = \kappa (\mathbf{X} - \boldsymbol{\chi}(\mathbf{X}, t)) - \eta \mathbf{U}(\mathbf{X}, t). \quad (6)$$

A separate line of research is focused on how to enforce these types of constraints exactly [64, 65], but doing so requires solving a coupled system of equations involving an exact Lagrange multiplier force along with the Eulerian velocity and pressure fields. The application of these types of approaches to the current sharp interface scheme is not addressed here.

2.2 Physical jump conditions

Throughout this subsection, \mathbf{X} and \mathbf{x} are taken to be corresponding positions in the reference and current configuration at time t , so that $\mathbf{x} = \boldsymbol{\chi}(\mathbf{X}, t)$. A jump in a scalar field ψ at position $\mathbf{x} = \boldsymbol{\chi}(\mathbf{X}, t)$ along the interface is

$$[\![\psi(\mathbf{x}, t)]\!] = \lim_{\epsilon \rightarrow 0^+} \psi(\mathbf{x} + \epsilon \mathbf{n}(\mathbf{x}, t), t) - \lim_{\epsilon \rightarrow 0^-} \psi(\mathbf{x} - \epsilon \mathbf{n}(\mathbf{x}, t), t) = \psi^+(\mathbf{x}, t) - \psi^-(\mathbf{x}, t), \quad (7)$$

in which $[\![\cdot]\!]$ indicates the jump value, $\mathbf{n}(\mathbf{x}, t)$ is the outward unit normal vector along the interface Γ_t in the current configuration, and $\psi^+(\mathbf{x}, t)$ and $\psi^-(\mathbf{x}, t)$ are the limiting values as approaching the interface position \mathbf{x} from the exterior region Ω_t^+ and interior region Ω_t^- , respectively. This definition can be extended for jumps in vectors in a component-wise manner.

For an incompressible Newtonian fluid, the fluid stress is

$$\bar{\bar{\boldsymbol{\sigma}}}_f(\mathbf{x}, t) = -p(\mathbf{x}, t) \mathbb{I} + \mu (\nabla \mathbf{u}(\mathbf{x}, t) + \nabla \mathbf{u}^T(\mathbf{x}, t)). \quad (8)$$

Because the boundary is infinitesimally thin, the interfacial force is balanced by a discontinuity in the fluid traction $\boldsymbol{\tau}_f = \bar{\bar{\boldsymbol{\sigma}}}_f(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t)$ along the interface,

$$j^{-1}(\mathbf{X}, t) \mathbf{F}(\mathbf{X}, t) = [\![\boldsymbol{\tau}_f(\mathbf{x}, t)]\!] = \boldsymbol{\tau}_f^+(\mathbf{x}, t) - \boldsymbol{\tau}_f^-(\mathbf{x}, t), \quad (9)$$

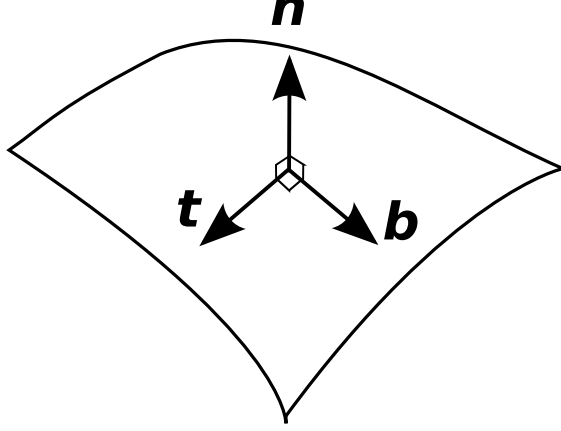


Figure 2: Unit normal (\mathbf{n}) and tangent (\mathbf{t} and \mathbf{b}) vectors on the interface.

in which j is the Jacobian determinant that converts the surface force density from force per unit area in the reference configuration to force per unit area in the current configuration. (The calculation of j is detailed in Sec. 3.3.)

Both velocity and pressure distributions are affected by the forces imposed along the immersed boundary. Away from the interface, these quantities can be assumed to be smooth, with the only possible discontinuities occurring at the immersed boundary. The no-slip and no-penetration conditions require the velocity at the immersed interface to be continuous,

$$\llbracket \mathbf{u}(\mathbf{x}, t) \rrbracket = 0. \quad (10)$$

Taking the dot-product of both sides in Eq. (8) by $\mathbf{n}(\mathbf{x}, t)$, and using Eq. (9), it is straightforward to show [48, 33]

$$\llbracket p(\mathbf{x}, t) \rrbracket = -j^{-1}(\mathbf{X}, t) \mathbf{F}(\mathbf{X}, t) \cdot \mathbf{n}(\mathbf{x}, t). \quad (11)$$

Jump conditions for the shear stress can be derived by taking the dot-product of Eq. (8) with tangential directions $\mathbf{b}(\mathbf{x}, t)$ and $\mathbf{t}(\mathbf{x}, t)$, see Fig. 2. If $\mathbf{n} = (n^x, n^y, n^z)$, $\mathbf{b} = (b^x, b^y, b^z)$, and $\mathbf{t} = (t^x, t^y, t^z)$ are the components of the local vectors in a local Cartesian coordinate system aligned with the interface, then

$$\begin{bmatrix} \mu \frac{\partial \mathbf{u}}{\partial x}(\mathbf{x}, t) \\ \mu \frac{\partial \mathbf{u}}{\partial y}(\mathbf{x}, t) \\ \mu \frac{\partial \mathbf{u}}{\partial z}(\mathbf{x}, t) \end{bmatrix} = \begin{pmatrix} n^x & b^x & t^x \\ n^y & b^y & t^y \\ n^z & b^z & t^z \end{pmatrix} \begin{bmatrix} \mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}}(\mathbf{x}, t) \\ \mu \frac{\partial \mathbf{u}}{\partial \mathbf{b}}(\mathbf{x}, t) \\ \mu \frac{\partial \mathbf{u}}{\partial \mathbf{t}}(\mathbf{x}, t) \end{bmatrix}. \quad (12)$$

After some manipulation, the jump conditions become

$$\left\llbracket \mu \frac{\partial \mathbf{u}}{\partial x}(\mathbf{x}, t) \right\rrbracket = (\bar{\bar{\mathbf{I}}} - \mathbf{n}(\mathbf{x}, t) \otimes \mathbf{n}(\mathbf{x}, t)) j^{-1}(\mathbf{X}, t) \mathbf{F}(\mathbf{X}, t) n^x, \quad (13)$$

$$\left\llbracket \mu \frac{\partial \mathbf{u}}{\partial y}(\mathbf{x}, t) \right\rrbracket = (\bar{\bar{\mathbf{I}}} - \mathbf{n}(\mathbf{x}, t) \otimes \mathbf{n}(\mathbf{x}, t)) j^{-1}(\mathbf{X}, t) \mathbf{F}(\mathbf{X}, t) n^y, \quad (14)$$

$$\left\llbracket \mu \frac{\partial \mathbf{u}}{\partial z}(\mathbf{x}, t) \right\rrbracket = (\bar{\bar{\mathbf{I}}} - \mathbf{n}(\mathbf{x}, t) \otimes \mathbf{n}(\mathbf{x}, t)) j^{-1}(\mathbf{X}, t) \mathbf{F}(\mathbf{X}, t) n^z, \quad (15)$$

which are convenient to use in the numerical implementation. Together, these jump conditions for the pressure and velocity gradient state that the discontinuity in the fluid stress across the interface balances the force concentrated along the interface.

We remark that although there are generally discontinuities in the velocity gradient at the interface, $\nabla \cdot \mathbf{u}(\mathbf{x}, t) \equiv 0$ holds throughout Ω . This can be seen by noting that the incompressibility condition holds for both the interior and exterior fluids, i.e., $\nabla \cdot \mathbf{u}^\pm(\mathbf{x}, t) = 0$. Applying the jump operator to both sides of this relation yields $\llbracket \nabla \cdot \mathbf{u}(\mathbf{x}, t) \rrbracket = 0$. Similarly, because the trajectories of material points cannot cross the interface, there also are no jump conditions associated with the convective derivative $D\mathbf{u}/Dt(\mathbf{x}, t)$.

3 Numerical Methods

This section describes the numerical approach, including the discretizations of the Eulerian and Lagrangian fields. For simplicity, the numerical scheme is explained in two spatial dimensions. The extension of the method to three spatial dimensions is straightforward.

3.1 Eulerian finite difference approximation

A staggered-grid (MAC) discretization is used for the incompressible Navier-Stokes equations, which approximates the pressure at cell centers and the velocity and forcing terms at the edges (in two dimensions) or faces (in three dimensions) of the grid cells. Standard second-order accurate centered approximation schemes are used for the divergence, gradient, and Laplace operators. The discrete divergence of the velocity $\mathbf{D} \cdot \mathbf{u}$ is evaluated at the cell centers, whereas the discrete pressure gradient $\mathbf{G}p$ and the components of the discrete Laplacian of the velocity $L\mathbf{u}$ are evaluated at the cell edges (or, in three dimensions, faces). A staggered-grid version [66, 67] of the xsPPM7 variant [68] of the piecewise parabolic method (PPM) [69] is used to approximate the nonlinear advection terms. Previously described methods for physical boundary conditions [67, 8] are used along the boundaries of the computational domain Ω . In some tests, we use a locally refined Eulerian discretization approach described by Griffith [8] that employs Cartesian grid adaptive mesh refinement (AMR).

To determine the effects of the known jump conditions, we classify all Eulerian grid locations as *regular* or *irregular*. Regular points are those for which none of the associated finite difference stencils cross the immersed boundary. Because the solution is continuous away from the interface, the discretization at these points does not need to be modified. For irregular points, however, at least one of the finite difference stencils associated with that point will cross the immersed boundary, and because the solution is discontinuous across the interface, standard discretizations that do not account for these jumps lead to large errors that do not decrease under grid refinement. This motivates the application of the jump conditions through correction terms in the discretization.

Using a Taylor series expansion [32, 47], it can be shown that if the interface cuts between two Cartesian grid points at location \mathbf{x}_o , such that $x_{i,j,k} \leq x_o < x_{i+1,j,k}$, with $\mathbf{x}_{i,j,k} \in \Omega^-$ and $\mathbf{x}_{i+1,j,k} \in \Omega^+$, then for a piecewise differentiable quantity ψ we have, for instance,

$$\frac{\partial \psi}{\partial x}(\mathbf{x}_{i+\frac{1}{2},j,k}) = \frac{\psi_{i+1,j,k} - \psi_{i,j,k}}{\Delta x} + \frac{\text{sgn}\{n^x\}}{\Delta x} \sum_{m=0}^2 \frac{(d^+)^m}{m!} \left[\frac{\partial^m \psi}{\partial x^m} \right]_{\mathbf{x}_o} + O(\Delta x^2), \quad (16)$$

$$\frac{\partial^2 \psi}{\partial x^2}(\mathbf{x}_{i,j,k}) = \frac{\psi_{i+1,j,k} - 2\psi_{i,j,k} + \psi_{i-1,j,k}}{\Delta x^2} + \frac{\text{sgn}\{n^x\}}{\Delta x^2} \sum_{m=0}^3 \frac{(d^+)^m}{m!} \left[\frac{\partial^m \psi}{\partial x^m} \right]_{\mathbf{x}_o} + O(\Delta x^2), \quad (17)$$

in which Δx is the grid spacing in the x direction, $\psi_{i,j,k} = \psi(\mathbf{x}_{i,j,k})$, $d^+ = x_{i+1,j,k} - x_o > 0$, and n^x is the x -component of the normal vector $\mathbf{n} = (n^x, n^y, n^z)$ at the intersection point \mathbf{x}_o .

We now demonstrate how these corrections can be applied to irregular stencils of the pressure and the viscous terms in the x -component of the Navier-Stokes equations for the two-dimensional staggered mesh arrangements shown in Fig. 3. The extension to three spatial dimensions is straightforward. We consider only the lowest-order jump conditions, accounting for discontinuities in the pressure and the derivative of the velocity. Let $\mathbf{G}p = (G^x p, G^y p)$ be the discrete pressure gradient. The modified discretization including the correction term for the x -component of this vector, e.g. as shown in Fig. 3(a), is

$$(G^x p)_{i+\frac{1}{2},j} = \frac{p_{i+1,j} - p_{i,j}}{\Delta x} - \frac{\llbracket p(\mathbf{x}, t) \rrbracket}{\Delta x}. \quad (18)$$

Similarly we represent the finite difference approximation of the vector Laplacian of $\mathbf{u} = (u, v)$ by $L\mathbf{u} = (L^x u, L^y v)$, in which L^x and L^y are in fact the same differencing operator applied to different components of the velocity vector. The modified discretization of the $L^x u$, e.g. as shown in Fig. 3(b), is

$$\begin{aligned} (L^x u)_{i-\frac{1}{2},j} &= \frac{u_{i+\frac{1}{2},j} - 2u_{i-\frac{1}{2},j} + u_{i-\frac{3}{2},j}}{\Delta x^2} + \frac{u_{i-\frac{1}{2},j+1} - 2u_{i-\frac{1}{2},j} + u_{i-\frac{1}{2},j-1}}{\Delta y^2} \\ &\quad + \frac{h_\alpha}{\Delta x^2} \left[\frac{\partial \mathbf{u}}{\partial x} \right]_\alpha - \frac{h_\beta}{\Delta y^2} \left[\frac{\partial \mathbf{u}}{\partial y} \right]_\beta, \end{aligned} \quad (19)$$

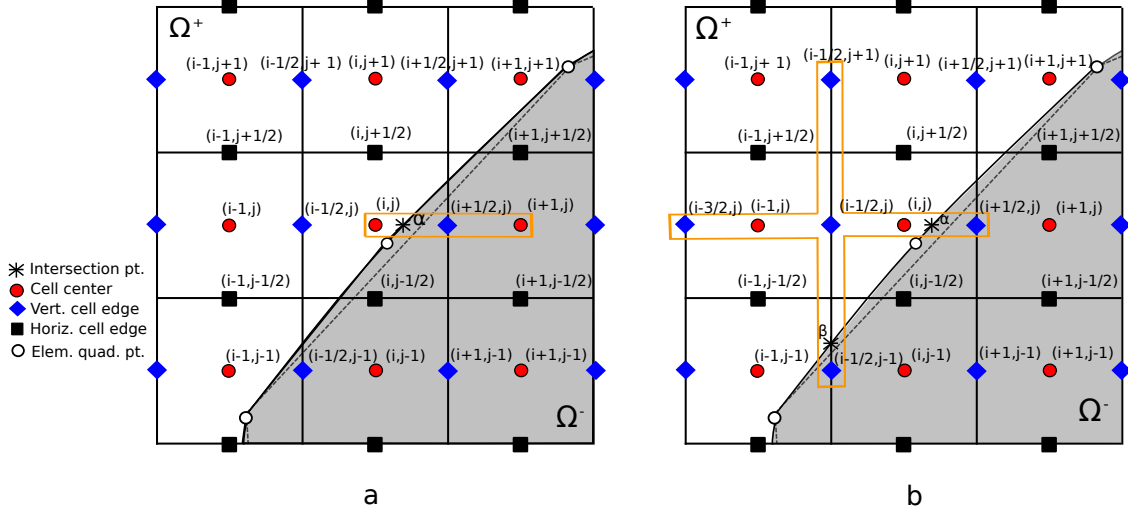


Figure 3: Two-dimensional illustration of irregular stencils in the Navier-Stokes equations for the x -component of (a) the pressure gradient $(G^x p)_{i+\frac{1}{2},j}$ and (b) Laplacian of the x -component of the velocity $(L^x u)_{i-\frac{1}{2},j}$

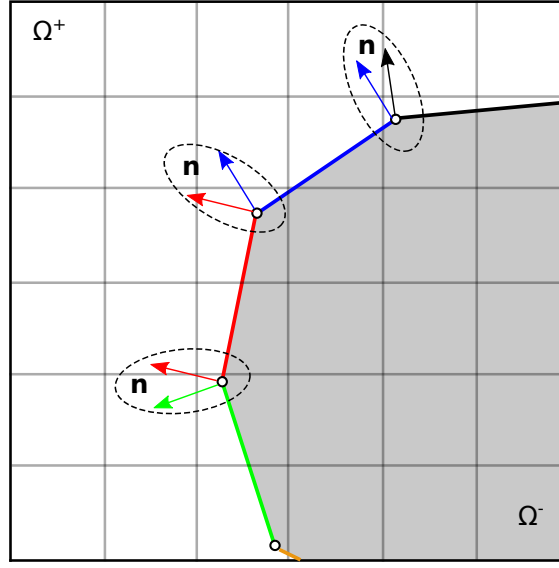


Figure 4: Two-dimensional schematic of a C^0 -continuous immersed boundary. Normal vectors are discontinuous at the junction between neighboring elements.

in which the positive values of h_α and h_β are defined as $h_\alpha = x_{i+\frac{1}{2},j} - x_\alpha$ and $h_\beta = y_\beta - y_{i-\frac{1}{2},j-1}$.

For the remainder of the paper, we assume an isotropic Cartesian grid, so that $\Delta x = \Delta y = \Delta z = h$.

3.2 Lagrangian discretization

To define a finite element approximation, consider \mathcal{T}_h as a triangulation of Γ_0 , the reference configuration of the interface, with elements U^e such that $\mathcal{T}_h = \cup_e U^e$, in which e indexes the mesh elements. $\{\mathbf{X}_l\}_{l=1}^M$ are the nodes of the mesh and $\{\phi_l(\mathbf{X})\}_{l=1}^M$ are the corresponding nodal (Lagrangian) basis functions. Using the finite element basis functions and the time-dependent physical positions of the nodes of the Lagrangian

mesh $\{\chi_l(t)\}_{l=1}^M$, the approximation to the interfacial deformation $\chi_h(\mathbf{X}, t)$ is defined by

$$\chi_h(\mathbf{X}, t) = \sum_{l=1}^M \chi_l(t) \phi_l(\mathbf{X}). \quad (20)$$

From this representation of the deformation, it is straightforward to evaluate approximations to geometrical quantities such as the surface normal or surface Jacobian determinant by differentiating Eq. (20). Because we use C^0 basis functions in the present work, quantities that are obtained in terms of $\frac{\partial \chi_h}{\partial \mathbf{X}}$ are discontinuous in both the reference and current configurations, as shown in Fig. 4.

3.3 Projected Jump conditions

Because the normal direction to the interface is generally discontinuous on a C^0 mesh at the mesh nodes, pointwise jump conditions determined from the mesh geometry and the surface Jacobian j are generally discontinuous, see Fig. 4. Instead of using these pointwise values directly, we project the jump conditions onto the subspace $S_h = \text{span}\{\phi_l(\mathbf{X})\}_{l=1}^M$. Given a function $\psi \in L^2(\Gamma_0)$, its L^2 projection $P_h\psi$ onto the subspace S_h is defined by requiring $P_h\psi$ to satisfy

$$\int_{\Gamma_0} (\psi(\mathbf{X}) - P_h\psi(\mathbf{X})) \phi_l(\mathbf{X}) dA = 0, \quad \forall l = 1, \dots, M. \quad (21)$$

The L^2 projection of a vector-valued quantity is determined component-wise. Because the L^2 projection is defined via integration, the function ψ does not need to be continuous or even to have well-defined nodal values. By construction, however, $P_h\psi$ will inherit any smoothness provided by the subspace S_h . In particular, for C^0 Lagrangian basis functions, $P_h\psi$ will be at least continuous. In our numerical scheme, we set $C_h^n(\mathbf{X}, t)$ to be the L^2 projection of the normal component of the surface force per unit current area, $j^{-1}\mathbf{F}_h(\mathbf{X}, t)$, onto S_h , and we set $\mathbf{C}_h^t(\mathbf{X}, t)$ to be the projection of the tangential component of the force. Solving for the projected jump conditions require solving linear systems of equations involving the mass matrix M with components $M_{kl} = \int \phi_k \phi_l dA$. In practice, Eq. (21) is approximated using a numerical quadrature rule. To simplify notation, the subscript “h” is mostly dropped in the remainder of the paper when showing numerical approximations to the Lagrangian variables

In implementing the jump conditions, it is convenient to evaluate them in the Cartesian directions. To do so, it is necessary to determine the normal and tangents to the interface. We let vectors \mathbf{e}_ι and \mathbf{e}_ν be tangents to the local element coordinates ι and ν , so that

$$\mathbf{e}_\iota = \begin{bmatrix} \frac{\partial x}{\partial \iota} \\ \frac{\partial y}{\partial \iota} \\ \frac{\partial z}{\partial \iota} \end{bmatrix} \quad \text{and} \quad \mathbf{e}_\nu = \begin{bmatrix} \frac{\partial x}{\partial \nu} \\ \frac{\partial y}{\partial \nu} \\ \frac{\partial z}{\partial \nu} \end{bmatrix}. \quad (22)$$

The derivatives of the global coordinates with respect to local coordinates can be determined using the basis functions via

$$\frac{\partial \chi}{\partial \iota}(\mathbf{X}, t) = \sum_{l=1}^M \chi_l(t) \frac{\partial \phi_l(\mathbf{X})}{\partial \iota} \quad \text{and} \quad \frac{\partial \chi}{\partial \nu}(\mathbf{X}, t) = \sum_{l=1}^M \chi_l(t) \frac{\partial \phi_l(\mathbf{X})}{\partial \nu}. \quad (23)$$

The *area-weighted* normal vector \mathbf{n} in the global coordinate is then obtained by evaluating $\mathbf{n} = \mathbf{e}_\iota \times \mathbf{e}_\nu$. Similarly for the normal in the local coordinate, \mathbf{N} , we use the Lagrangian material coordinates,

$$\mathbf{E}_\iota = \begin{bmatrix} \frac{\partial X}{\partial \iota} \\ \frac{\partial Y}{\partial \iota} \\ \frac{\partial Z}{\partial \iota} \end{bmatrix} \quad \text{and} \quad \mathbf{E}_\nu = \begin{bmatrix} \frac{\partial X}{\partial \nu} \\ \frac{\partial Y}{\partial \nu} \\ \frac{\partial Z}{\partial \nu} \end{bmatrix}, \quad (24)$$

yielding $\mathbf{N} = \mathbf{E}_\iota \times \mathbf{E}_\nu$. The surface Jacobian, which is the area ratio between the two coordinates systems, is obtained via

$$j = \frac{dA}{da} = \frac{\|\mathbf{N}\|}{\|\mathbf{n}\|}. \quad (25)$$

In the present study, $j \approx 1$ because we allow for only infinitesimal deformations of the interface.

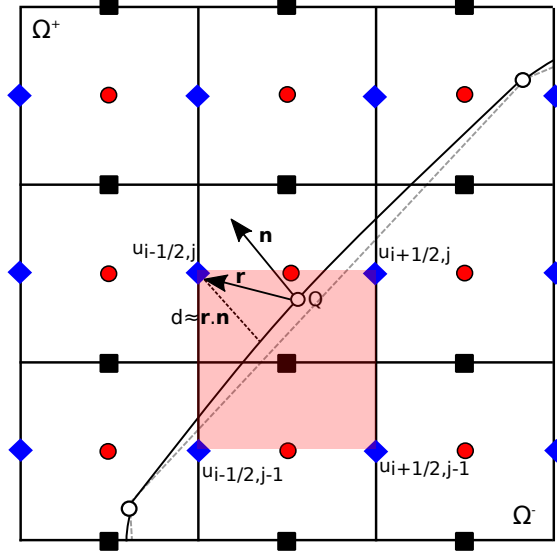


Figure 5: Two-dimensional schematic of the interpolation scheme for the x -component of the velocity at the interfacial location $\mathbf{x} = \chi(\mathbf{X}, t)$. The cell used for calculating the modified bilinear interpolation stencil is highlighted in red.

3.4 Intersection algorithm

Intersections between the Cartesian finite difference stencils and the interface are determined by a ray-casting algorithm. In two dimensions, this is done by finding the intersections of Cartesian-oriented lines with first-order parametric elements, which amounts to solving a single linear equation. In three-dimensional cases, we use the Möller-Trumbore ray-triangle intersection algorithm, which is a fast method to calculate intersections with surface triangles [70], and a similar efficient approach for convex planar quadrilateral is used to find intersections of stencils with rectangular surface elements [71]. Although there should be at most one intersection between a given finite difference stencil and the interface, finite precision effects may yield spurious multiple intersections, or may completely miss intersections between the finite difference stencil and the interface. To avoid these situations, before determining these intersections, we perturb the positions of the control points of the interface mesh away from the cell centers, nodes, edges, and faces of the background Cartesian grid by a factor proportional to $\sqrt{\epsilon_{\text{mach}}}$, in which ϵ_{mach} is machine precision. This is to ensure that we do not have to use specialized methods for treating apparent multiple intersections. This perturbation in the position of the interface is much smaller than the accuracy of the overall method, and it avoids the need to implement specialized discrete geometry algorithms that explicitly treat such spurious intersections.

3.5 Velocity interpolation

The goal here is to determine a velocity-restriction operator $\mathcal{J} = \mathcal{J}[\chi, \mathbf{F}]$ to determine the Lagrangian mesh velocity \mathbf{U} from the Eulerian grid velocity \mathbf{u} . The velocity interpolation used herein consists of two steps. Ultimately, we project the interpolated velocity onto the Lagrangian basis functions to determine the motion of the interface, and so the first step of the interpolation procedure is to evaluate the Lagrangian velocity \mathbf{V} at the quadrature points along the interface that are used to compute the L^2 projection of the interpolated velocity. The second step projects the velocity field \mathbf{V} onto the space S_h spanned by the nodal finite element basis functions to determine the mesh motion \mathbf{U} .

Because the projection step is standard, we focus here on evaluating \mathbf{V} at a generic reference position \mathbf{X} on the interface, with corresponding physical position $\mathbf{x} = \chi(\mathbf{X}, t)$. Our approach is to use modified bilinear (or, in three spatial dimensions, trilinear) interpolation that accounts for the known discontinuities in the velocity gradient at the interface. Considering the two-dimensional schematic detailed in Fig. 5, the general

formula for the x -component of the velocity, V^x , at the interfacial location is

$$V^x(\mathbf{X}, t) = (1 - \zeta)(1 - \lambda)u_{i-\frac{1}{2}, j-1} + C_{i-\frac{1}{2}, j-1} + \zeta(1 - \lambda)u_{i+\frac{1}{2}, j-1} + C_{i+\frac{1}{2}, j-1} \\ + \zeta\lambda u_{i+\frac{1}{2}, j} + C_{i+\frac{1}{2}, j} + (1 - \zeta)\lambda u_{i-\frac{1}{2}, j} + C_{i-\frac{1}{2}, j}, \quad (26)$$

in which $\lambda = \frac{x - x_{i-\frac{1}{2}, j-1}}{h}$ and $\zeta = \frac{y - y_{i-\frac{1}{2}, j-1}}{h}$ and $\Delta x = \Delta y = h$ is the (isotropic) Cartesian grid spacing. The corrections $C_{i-\frac{1}{2}, j-1}$, $C_{i+\frac{1}{2}, j-1}$, $C_{i+\frac{1}{2}, j}$ and $C_{i-\frac{1}{2}, j}$ have the following forms:

$$C_{i-\frac{1}{2}, j-1} = \begin{cases} h(1 - \zeta)(1 - \lambda)(\zeta n^x \llbracket \frac{\partial u}{\partial x} \rrbracket + \lambda n^y \llbracket \frac{\partial u}{\partial y} \rrbracket) & \mathbf{x}_{i-\frac{1}{2}, j-1} \in \Omega^+, \\ 0 & \mathbf{x}_{i-\frac{1}{2}, j-1} \in \Omega^-, \end{cases} \quad (27)$$

$$C_{i+\frac{1}{2}, j-1} = \begin{cases} -h\zeta(1 - \lambda)((1 - \zeta)n^x \llbracket \frac{\partial u}{\partial x} \rrbracket - \lambda n^y \llbracket \frac{\partial u}{\partial y} \rrbracket) & \mathbf{x}_{i+\frac{1}{2}, j-1} \in \Omega^+, \\ 0 & \mathbf{x}_{i+\frac{1}{2}, j-1} \in \Omega^-, \end{cases} \quad (28)$$

$$C_{i+\frac{1}{2}, j} = \begin{cases} -h\zeta\lambda((1 - \zeta)n^x \llbracket \frac{\partial u}{\partial x} \rrbracket + (1 - \lambda)n^y \llbracket \frac{\partial u}{\partial y} \rrbracket) & \mathbf{x}_{i+\frac{1}{2}, j} \in \Omega^+, \\ 0 & \mathbf{x}_{i+\frac{1}{2}, j} \in \Omega^-, \end{cases} \quad (29)$$

$$C_{i-\frac{1}{2}, j} = \begin{cases} h(1 - \zeta)\lambda(\zeta n^x \llbracket \frac{\partial u}{\partial x} \rrbracket - (1 - \lambda)n^y \llbracket \frac{\partial u}{\partial y} \rrbracket) & \mathbf{x}_{i-\frac{1}{2}, j} \in \Omega^+, \\ 0 & \mathbf{x}_{i-\frac{1}{2}, j} \in \Omega^-. \end{cases} \quad (30)$$

Notice that the jump conditions for the velocity gradient can be determined from the interfacial force \mathbf{F} . This modified bilinear interpolation is similar to the approach of Tan et al. [72, 73]. An important difference is in the value used for the distance from the Cartesian grid locations to the interface in the Taylor series expansion of the correction terms. The approach by Tan et al. uses the distance between the grid points and the interfacial point ($|\mathbf{r}|$ in Fig. 5). Here, this is modified by projecting the location vector \mathbf{r} to the normal vector \mathbf{n} and using $|\mathbf{r} \cdot \mathbf{n}|$ instead. Taking into account for the normal vector naturally provides a more accurate approximation to the actual distance between the grid point and the interface.

$\mathbf{V}(\mathbf{X}, t)$ can be evaluated at any $\mathbf{X} \in \Gamma_0$, but generally, $\mathbf{V}(\mathbf{X}, t) \notin S_h = \text{span}\{\phi_l(\mathbf{X})\}$. Therefore, the second step of the interpolation is to set $\mathbf{U} = P_h \mathbf{V}$, component-by-component. This completes the construction of $\mathbf{U} = \mathcal{J}[\chi, \mathbf{F}] \mathbf{u}$.

3.6 Calculating the fluid traction

Tests of the present method will evaluate components of the fluid traction along the interface. For simplicity, we focus on evaluating exterior tractions, but the evaluation of interior tractions is similar. To obtain an approximation to the exterior pressure at position \mathbf{x} along the interface, we use

$$p_h^+(\mathbf{x}, t) = \llbracket p(\mathbf{x}, t) \rrbracket + \mathcal{I}[p](\mathbf{x}^-, t) \quad (31)$$

in which $p^- = \mathcal{I}[p](\mathbf{x}^-, t)$ is the interior pressure interpolated to a position \mathbf{x}^- away from the interface in the opposite direction of the normal vector \mathbf{n} and at a distance equal to 1.2 times the diagonal size of one grid cell. Here, \mathcal{I} is the unmodified bilinear (or trilinear) interpolation operator. By construction, \mathbf{x}^- is sufficiently far from the interface to ensure that uncorrected interpolation may be used without degradation in accuracy. To compute the exterior wall shear stress, the one-sided normal derivative of the velocity is calculated using the interfacial velocity reconstruction described in Sec. 3.5 along with the velocity value at a neighboring location in the direction of the normal vector \mathbf{x}^+ . As with the pressure, unmodified bilinear (or trilinear) interpolation is used to obtain the velocity away from the interface. A one-sided finite difference formula is used to obtain an approximation to the normal derivative as

$$\left(\frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right)_h^+ (\mathbf{x}, t) = \frac{\mathcal{I}[\mathbf{u}](\mathbf{x}^+, t) - \mathbf{u}(\mathbf{x}, t)}{\hat{h}}. \quad (32)$$

where the distance \hat{h} is chosen to be slightly larger than the diagonal size of the Cartesian mesh (1.05 times the diagonal size), so that regular bilinear (or trilinear) interpolation can be used to evaluate $\mathcal{I}[\mathbf{u}](\mathbf{x}^+, t)$. A second-order formula using a three-point stencil can also be used that requires interpolating an additional

point in the normal direction. However, preliminary numerical experiments (data not shown) suggest the computation using only two points suppresses oscillations that appear to be associated with the higher-order stencil. Moreover, as shown in our tests, this simple scheme is adequate to achieve the designed first-order accuracy of the wall shear stress.

Note that as with velocity interpolation, the pressure and wall shear stress can be evaluated at arbitrary locations along the interface. As before, to obtain nodal values of these quantities along the interface, we use the L^2 projection of the interfacial values.

3.7 Time-integration scheme

Starting from the values of χ^n and \mathbf{u}^n at time t^n and $p^{n-\frac{1}{2}}$ at time $t^{n-\frac{1}{2}}$, we must compute χ^{n+1} , \mathbf{u}^{n+1} , and $p^{n+\frac{1}{2}}$. To do this, we first determine a preliminary approximation to the structure location at time $t^{n+\frac{1}{2}}$ by

$$\frac{\hat{\chi}^{n+1} - \chi^n}{\Delta t} = \mathbf{U}^n = \mathcal{J}^n \mathbf{u}^n \quad (33)$$

with $\mathcal{J}^n = \mathcal{J}[\chi^n, \mathbf{F}^n]$ is the discrete velocity restriction operator described in Sec. 3.5. We also define an approximation to χ at time $t^{n+\frac{1}{2}}$ by

$$\chi^{n+\frac{1}{2}} = \frac{\hat{\chi}^{n+1} + \chi^n}{2}. \quad (34)$$

Next, we solve for χ^{n+1} , \mathbf{u}^{n+1} , and $p^{n+\frac{1}{2}}$ via

$$\rho \left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{A}^{n+\frac{1}{2}} \right) = -Gp^{n+\frac{1}{2}} + \mu \mathbf{L} \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) + \mathbf{f}^{n+\frac{1}{2}}, \quad (35)$$

$$\mathbf{D} \cdot \mathbf{u}^{n+1} = 0, \quad (36)$$

$$\frac{\chi^{n+1} - \chi^n}{\Delta t} = \mathbf{U}^{n+\frac{1}{2}} = \mathcal{J}^{n+\frac{1}{2}} \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right), \quad (37)$$

in which \mathbf{G} , $\mathbf{D} \cdot$, and \mathbf{L} are, respectively, the discrete gradient, divergence, and Laplace operators, $\mathbf{A}^{n+\frac{1}{2}} = \frac{3}{2}\mathbf{A}^n - \frac{1}{2}\mathbf{A}^{n-1}$ is obtained from a high-order upwind spatial discretization of the nonlinear convective term $\mathbf{u} \cdot \nabla \mathbf{u}$ [67], and $\mathbf{f}^{n+\frac{1}{2}}$ is discrete Eulerian body force on the Cartesian grid that corresponds to the sum of all the correction terms computed using the Lagrangian force $\mathbf{F} = \mathbf{F}[\chi^{n+\frac{1}{2}}, \mathbf{U}^n, t^{n+\frac{1}{2}}]$. The semi-implicit Crank-Nicolson form in Eq. (35) requires only linear solvers for the time-dependent incompressible Stokes equations. This system of equations is solved via the flexible GMRES (FGMRES) algorithm with an approximate factorization preconditioner based on the projection method that uses inexact subdomain solvers [67]. In the initial time step, a two-step predictor-corrector method is used to determine the velocity, deformation, and pressure, see Griffith and Luo [74] for further details. Note that similar to the standard IB method, we can relate \mathbf{f} and \mathbf{F} by a force-spreading operator $\mathcal{S} = \mathcal{S}[\chi]$ such that $\mathbf{f} = \mathcal{S}[\chi] \mathbf{F}$. Unlike the conventional IB method, however, the present force-spreading and velocity-restriction operators are not adjoints.

3.8 Standard IB formulation

For tests in this study that use the conventional IB method, we approximate the force using the nodal forces $\{\mathbf{F}_l(t)\}$ and the shape functions, so that

$$\mathbf{F}_h(\mathbf{X}, t) = \sum_{l=1}^M \mathbf{F}_l(t) \phi_l(\mathbf{X}). \quad (38)$$

Using Eq. (5), the discretized Lagrangian penalty force $\mathbf{F}_h(\mathbf{X}, t)$ can then be directly evaluated from χ_h and the interface motion \mathbf{U}_h . These forces are spread to the background grid using a discretized integral transform with a regularized delta function kernel. The adjoint of this operator is used to determine the velocity of the mesh from the Cartesian grid velocity field. In the present study, we use a piecewise linear regularized delta function. Details are provided by Griffith and Luo [74].

3.9 Software implementation

The present approach is implemented in the open-source IBAMR software [75], which is a C++ framework for FSI modeling using the IB formulation. IBAMR provides support for large-scale simulations through the use of distributed-memory parallelism and adaptive mesh refinement (AMR). IBAMR relies upon other open-source software libraries, including SAMRAI [76, 77], PETSc [78, 79, 80], *hypra* [81, 82], and *libMesh* [83, 84], for various key functionalities.

4 Numerical results

Verification examples in two and three spatial dimensions, including comparisons to benchmark computational and experimental studies, are used to investigate the accuracy of the proposed IIM. We consider tests involving internal and external flow conditions, treating both stationary boundaries and boundaries with prescribed kinematics. All the interfaces in two and three-dimensions are covered by linear (P^1) Lagrangian elements. Wherever there is an exact solution to compare against, the analytic interface geometry is used to determine the exact solution. Those computations use only the discrete interface geometry, which in our tests are described using either one-dimensional line elements (in two spatial dimensions) or two-dimensional triangular or quadrilateral surface elements (in three spatial dimensions). Analytic information about the interface geometry is only used to determine the discrete representation of the reference configuration of the interface. When assessing the accuracy of the computed fluid traction, we always consider the traction on the side of the interface with the nontrivial flow.

In all cases, the penalty forces that approximately impose the interface kinematics act to support physical forces exerted by the flow on the interface, and the physical forces are a property of the model. Thus, under grid refinement, we expect that the penalty forces will converge to the physical loading forces. Notice also that the penalty forces are proportional to the displacement $\boldsymbol{\xi}(\mathbf{X}, t) - \boldsymbol{\chi}(\mathbf{X}, t)$ between the prescribed and computed interface positions. If we wish to achieve $\|\boldsymbol{\xi}(\mathbf{X}, t) - \boldsymbol{\chi}(\mathbf{X}, t)\| = O(h^2)$, it is necessary that the penalty parameter κ also satisfies $\kappa = O(1/h^2)$, so that an applied penalty force of the form $\mathbf{F} = \kappa(\boldsymbol{\xi}(\mathbf{X}, t) - \boldsymbol{\chi}(\mathbf{X}, t))$ satisfies $\|\mathbf{F}\| = O(1)$ under grid refinement. Because we maintain $\Delta t = O(h)$ in our convergence tests, which keeps the advective Courant-Friedrichs-Lewy (CFL) number fixed under grid refinement, it is convenient to choose $\kappa = \kappa_0/\Delta t^2$. We choose κ_0 by, first, choosing the finest grid spacing to be considered in the convergence study, say h_{\min} , along with the corresponding time step size Δt_{\min} . We then empirically determine approximately the largest value of κ_0 that allows the scheme to remain stable when $h = h_{\min}$ and $\Delta t = \Delta t_{\min}$. We then use the prescribed relationship between κ and Δt to determine κ for all coarser cases. This allows us to ensure that the numerical parameters are stable for all grid spacings considered in each test while using scalings that ensure that the method achieves its designed order of accuracy. All computations use a tight relative convergence threshold of $\epsilon_{\text{rel}} = 1\text{e-}10$ for all iterative linear solvers.

When assessing the accuracy of discontinuous quantities, such as the Eulerian pressure p , small perturbations in the interface position can result in $O(1)$ differences in the computed value at spatial locations \mathbf{x} that are close to the interface, even if the interface position is computed very accurately. At points within a mesh width of the interface, the pointwise errors can differ by the value of the jump in the solution at the interface. These geometrical errors can result from approximation error as well as the finite precision effects. To assess the impact of these unavoidable geometrical effects on the accuracy of the computed solutions, we report pointwise errors in the pressure both over the full domain Ω , and over the subregion Ω^* that excludes a band with a width of two grid cells around the interface. Pointwise errors in the quantity denoted p^* are computed over Ω^* . The numerical method is also shown to capture discontinuous quantities such as the pressure and wall shear stress on the interface itself, which indicates that errors in the location of the interface that effect the pointwise values of the Eulerian fields do not spoil the ability of the method to impose jump conditions accurately on those fields.

4.1 Pressure-driven flow

The first example considers steady-state pressure-driven flow inside a channel. Results are presented for both two-dimensional plane-Poiseuille channel flow as well as three-dimensional Hagen-Poiseuille flow inside

a circular pipe. In each case, both grid-aligned and skew structure configurations are used. For these tests, the interface comes into contact with the exterior boundary of the computational domain, $\partial\Omega$, and errors at these junctions can dominate some pointwise errors. Because we wish to examine the accuracy of the method for fully developed flows, and not to assess the treatment of numerical boundary conditions for the flow solver, we perform our error analyses over a slightly smaller region that excludes positions near $\partial\Omega$ that are within 10% of the length of the computational domain.

4.1.1 Plane-Poiseuille flow

We first consider a two-dimensional domain $\Omega = [0, L]^2$. A horizontal channel of width H extends across the middle of the domain. The two channel ends at $x = 0$ and $x = L$ are subject to constant pressure p_0 and $-p_0$ boundary conditions, establishing a constant pressure gradient of $2p_0/L$ across the channel. The prescribed-pressure boundary condition is imposed by means of applying combined normal-traction and zero-tangential-slip boundary conditions. At the remaining parts of the left ($x = 0$) and right ($x = L$) sides of Ω along with the entire bottom ($y = 0$) and top ($y = L$) boundaries, solid wall (zero-velocity) boundary conditions are imposed. The channel walls are modeled using one-dimensional linear (P^1) Lagrangian elements with an element size twice that of the Eulerian grid spacing, which results in a mesh ratio of $M_{\text{fac}} = 2$. Note that in this context and throughout the rest of this paper, M_{fac} is the ratio of Lagrangian element size to the Eulerian grid spacing. The steady-state solution of pressure-driven flow for such a channel is described by the plane Poiseuille equation,

$$u(y) = \frac{p_0 H}{\mu L} (y - y_0) \left(1 - \frac{y - y_0}{H} \right), \quad (39)$$

$$v = 0, \quad (40)$$

$$p(x) = p_0 - 2p_0 x/L, \quad (41)$$

in which y_0 is the height of the lower wall of the channel. Remaining simulation parameters include $\rho = 1$, $\mu = 0.01$, $L = 5$, $H = 1$, $y_0 = 2$ and $p_0 = 0.2$, resulting in a maximum velocity of $U_{\text{max}} = 1$, an average velocity of $\bar{U} = 2/3$, and the Reynolds number $\text{Re} = \frac{\rho H \bar{U}}{\mu} \approx 66.66$. The time step size is $\Delta t = 0.1h$, yielding a maximum advective CFL number of approximately 0.1–0.2. The penalty parameters are $\kappa = 10^{-3}/\Delta t^2$ and $\eta = 0$.

Along with the horizontal plane-Poiseuille flow, an inclined version of the channel is also studied in which the immersed boundary is not aligned with the Cartesian directions. In this slightly more challenging scenario, the channel is at an angle $\theta = \pi/12$ with the horizontal direction. At the inlet and outlet of the channel, the rotated exact solution is used to determine normal traction and tangential velocity boundary conditions, so that the flow conditions are the same as in the aligned configuration. At the remaining parts of $\partial\Omega$, solid-wall boundary conditions are imposed. Other than the rotation in the geometry, all other simulation parameters are the same as the horizontal case.

To provide a qualitative comparison of the present immersed interface method to the conventional IB method, Fig. 6 shows the velocity and pressure in the horizontal channel for a relatively coarse grid spacing of $h = 0.156$ using both numerical approaches. Fig. 7 is similar, but considers the skew configuration at a finer grid resolution of $h = 0.078$. As compared to the IB method, the present immersed interface scheme yields sharply resolved pressure and velocity fields.

To further analyze the improvement in accuracy offered by jump condition-based discretization approaches, Fig. 8 shows the incremental changes in the accuracy of the scheme as the jump conditions are systematically incorporated. The horizontal channel with a grid spacing of $h = 0.078$ is considered, and pressure and velocity profiles are plotted at $x = 3.5$. In the figure, *Step 1* indicates results obtained by a simplified version of the present method in which only the normal component of the interfacial force is used to determine pressure jump conditions. The tangential component of the force is transmitted to the background grid using a standard IB approach with a piecewise linear kernel function. In addition, the velocity is also interpolated to the interface using the conventional IB approach with a piecewise linear kernel function. Thus, in Step 1, only pressure discontinuities are sharply resolved, and discontinuities in the velocity gradient are regularized. This scheme is similar to the IIM of Lee and LeVeque [33]. The result demonstrates that adding the correction due to the pressure discontinuity yields a significant improvement in the accuracy of

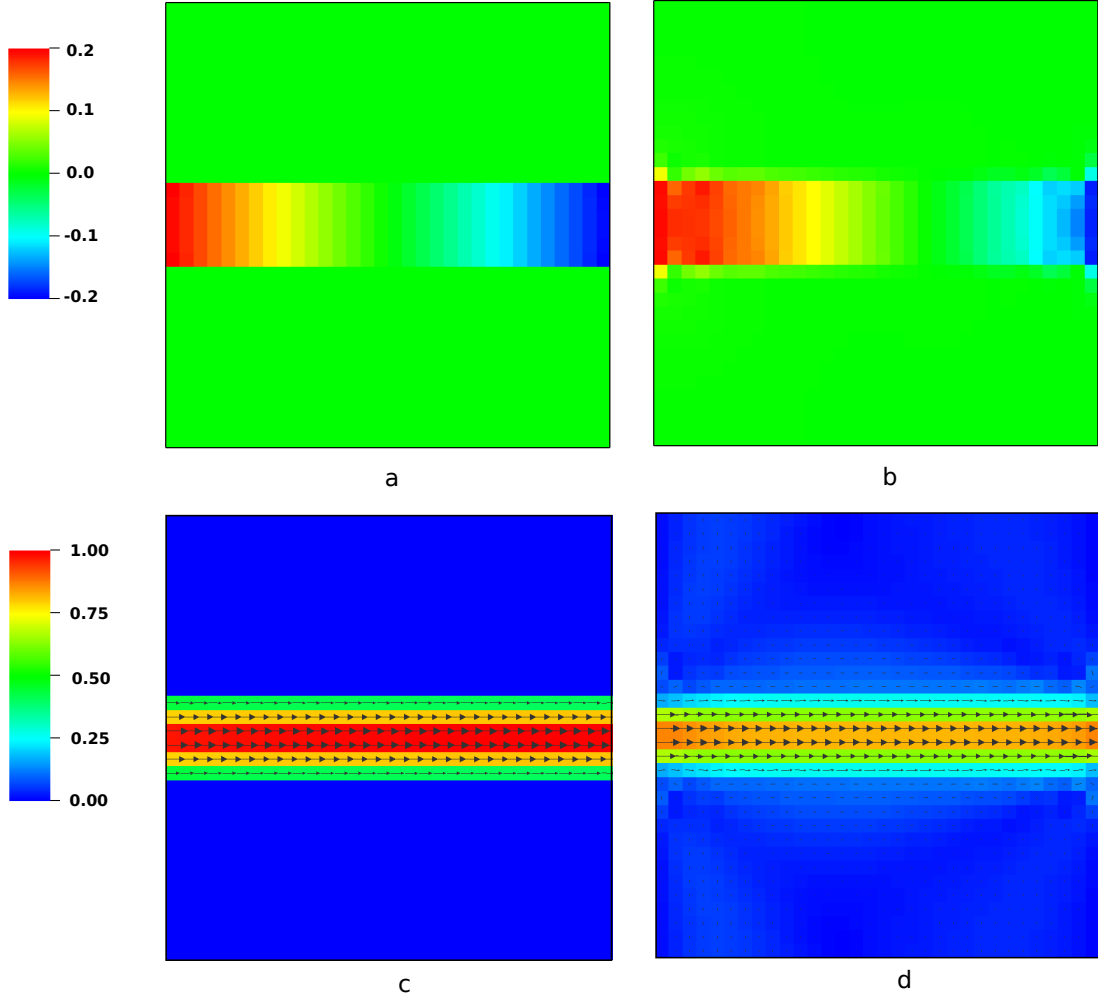


Figure 6: Comparison of the steady-state pressure (top panels) and velocity (bottom panels) distributions between the present IIM (left panels) and the standard IB method (right panels) for the horizontal channel with grid spacing $h = 0.156$. The IIM sharply resolves both the pressure and flow, and yields higher flow rates at comparable grid spacings than the IB method. See also Fig. 8.

the pressure as compared to the conventional IB method, but the velocity profile at the immersed boundary still suffers from low accuracy. *Step 2* indicates results obtained by a version of the simplified method that is modified so that the tangential portion of the force is also used to impose jump conditions in the velocity gradient along the interface, but the velocity is still interpolated to the interface using the conventional IB approach with a piecewise linear kernel function. Notice that there remains a sizable difference between the Step 2 velocity profile and the exact solution, which results from the use of uncorrected interpolation to determine the velocity of the interface. *Full Method* indicates results obtained by the full immersed interface method presented in this work. It is clear that if all jump conditions are included both in applying the forces and in interpolating the velocities, the resulting method is in excellent agreement with the analytical solutions for both velocity and pressure. The accuracy of the scheme is degraded by omitting any of these corrections.

Grid convergence studies are performed for both the horizontal and inclined configuration. The errors in the computed Eulerian velocity and pressure in L^2 and L^∞ norms are summarized in Fig. 9. These errors are also tabulated and reported in A. Second-order convergence of the velocity is observed in the L^2 norm for both the horizontal and inclined cases, and the global order of accuracy in the L^2 norm of the

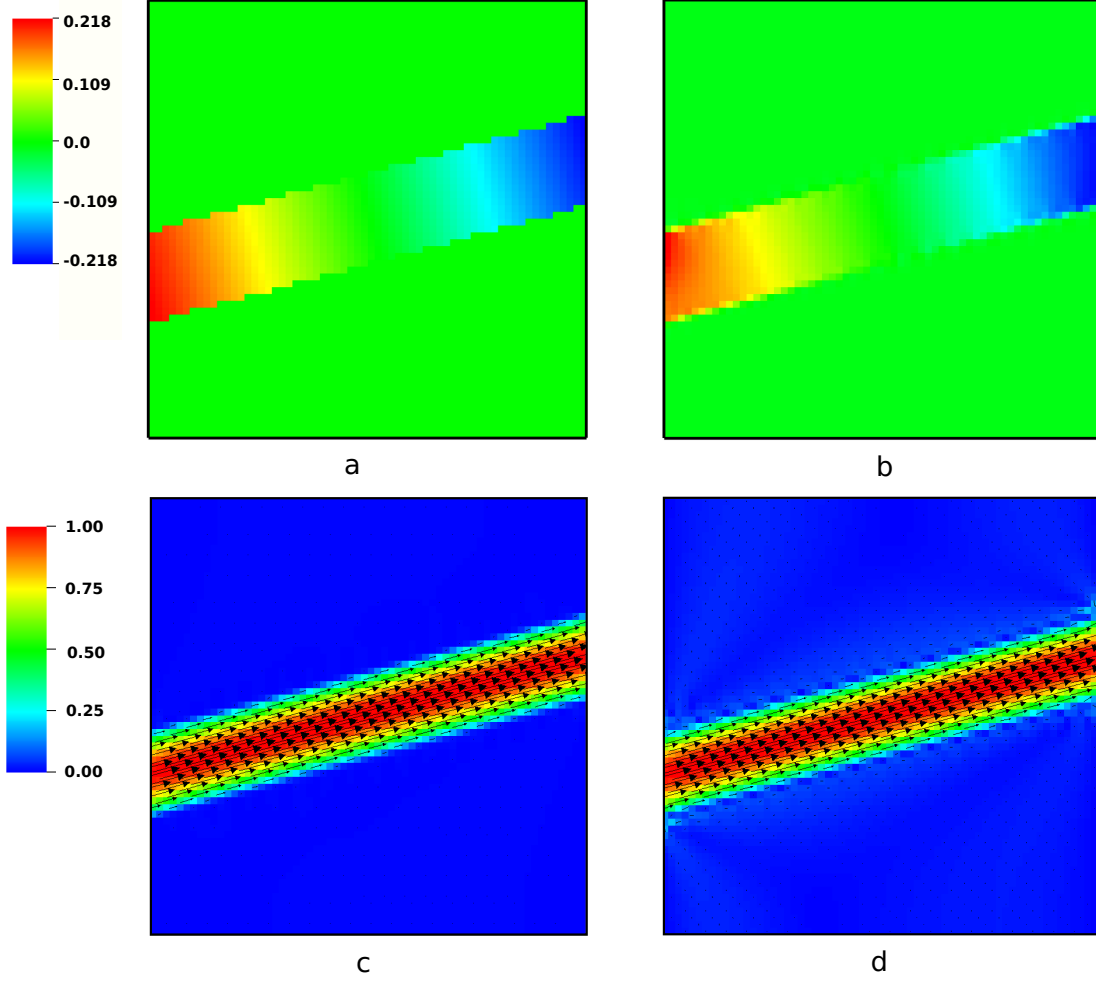


Figure 7: Comparison of the steady-state pressure (top panels) and velocity (bottom panels) distributions between the present IIM (left panels) and the standard IB method (right panels) for the inclined channel with grid spacing $h = 0.078$. The results are qualitatively similar to those of the horizontal case shown in Fig. 6, but note that here we are using a finer computational grid, and so both the IIM and IB results are more accurate than those of Fig. 6.

pressure appears to be between first- and second-order. As already discussed, errors and convergence rates for discontinuous quantities such as the pressure are sensitive to the precise location of the discontinuity, and small perturbations in the position of the interface can result in large errors when compared to the analytic solution obtained with the exact interfacial geometry. Consequently, we report L^∞ errors for both the full Eulerian pressure field p and also for the field p^* that is obtained by excluding grid cells within $2h$ of the interface. Although the convergence of p in the L^∞ norm is slow, p^* shows first-order pointwise convergence rates.

Fig. 10 reports the errors in the computed Lagrangian displacement, velocity, pressure, and wall shear stress, and tabulated errors are provided in A. At least second-order convergence is observed for the displacement in the L^2 and L^∞ norms and for the velocity in the L^2 norm. The L^∞ norm of the velocity error in the inclined channel test appears to converge at slightly less than second order. Also at least 1.5-order accuracy is achieved in the pressure in the L^2 and L^∞ norms along with first-order convergence in the wall shear stress in the L^2 and L^∞ norms.

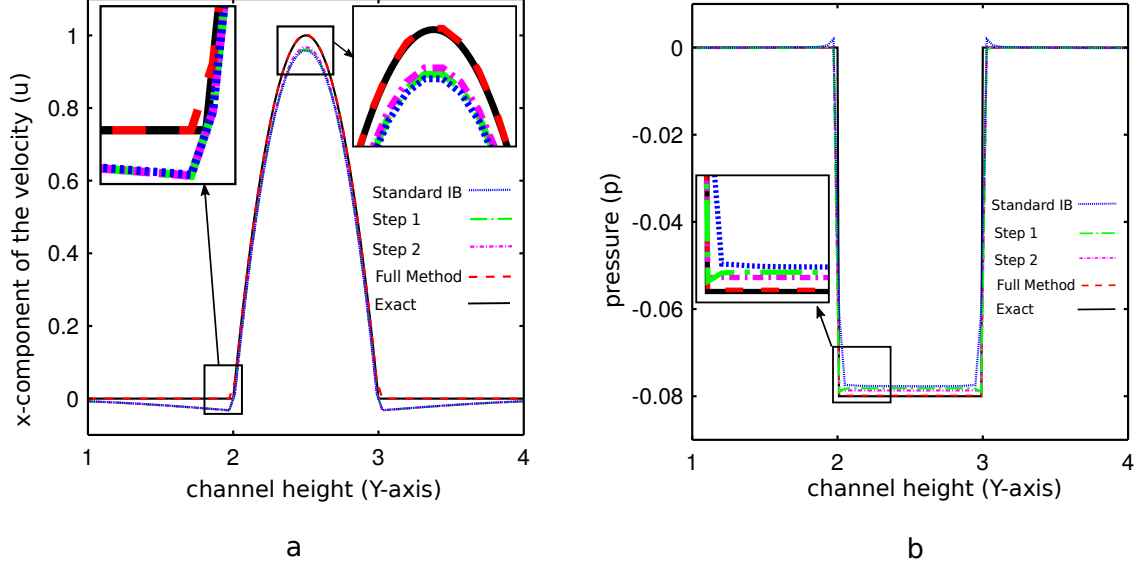


Figure 8: Comparisons of (a) velocity and (b) pressure profiles at $x = 3.5$ for the plane-Poiseuille flow with grid spacing $h = 0.078$. Here, *Standard IB* corresponds to a standard IB method with piecewise-linear regularized delta functions, *Step 1* corresponds to a hybrid IB-IIM that only imposes jump conditions for the pressure, *Step 2* corresponds to an IIM that imposes jump conditions for both the pressure and velocity gradient but uses uncorrected bilinear interpolation to determine the motion of the interface, and *Full Method* corresponds to the IIM described in this paper. Notice that the full method is substantially more accurate than the other methods considered here.

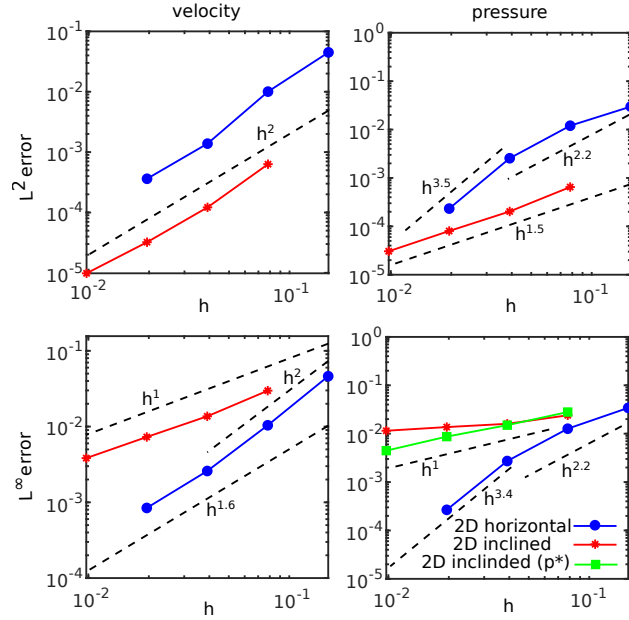


Figure 9: Convergence of the L^2 and L^∞ norms of the error of the Eulerian velocity and pressure for the horizontal and inclined plane-Poiseuille flows. Simulation parameters include $\text{Re} = 66.66$, $\Delta t = 0.1h$, and $M_{\text{fac}} = 2$.

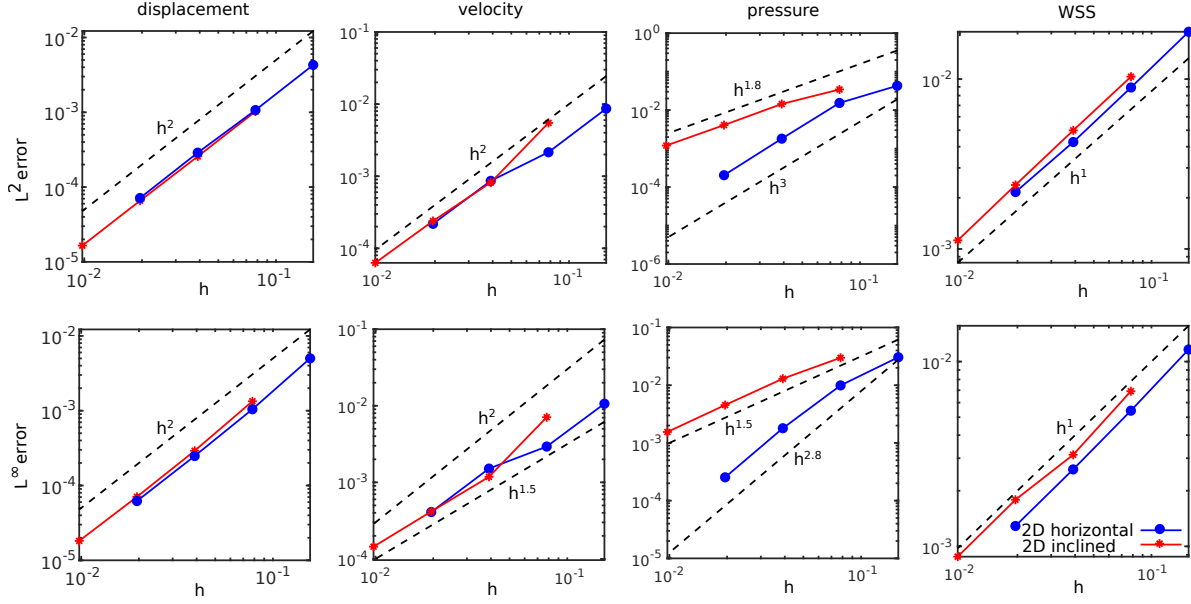


Figure 10: Convergence of the L^2 and L^∞ norms of the errors of the Lagrangian displacement, velocity, pressure, and wall shear stress (WSS) for the horizontal and inclined plane-Poiseuille flow, determined along the lower side of the channel. Simulation parameters include $Re = 66.66$, $\Delta t = 0.1h$, and $M_{fac} = 2$.

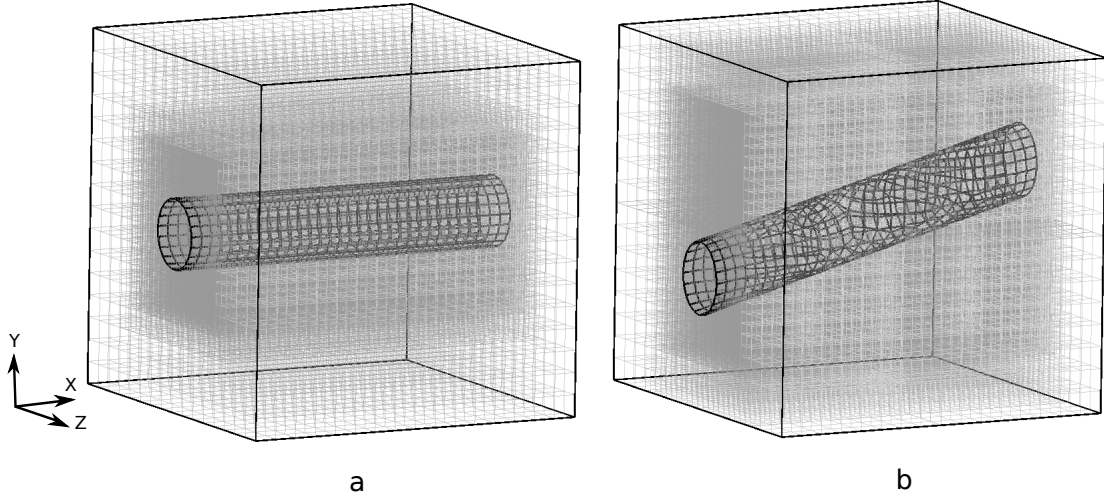


Figure 11: Computational meshes used to compute flow in (a) the horizontal pipe and (b) the inclined pipe. A locally refined Cartesian grid is used in both cases.

4.1.2 Hagen-Poiseuille flow

The steady-state solution for a three-dimensional axisymmetric Hagen-Poiseuille flow in the x -direction is

$$u(r) = \frac{p_0}{2\mu L} (R^2 - r^2), \quad (42)$$

$$v = 0, \quad (43)$$

$$w = 0, \quad (44)$$

$$p(x) = p_0 - 2p_0x/L, \quad (45)$$

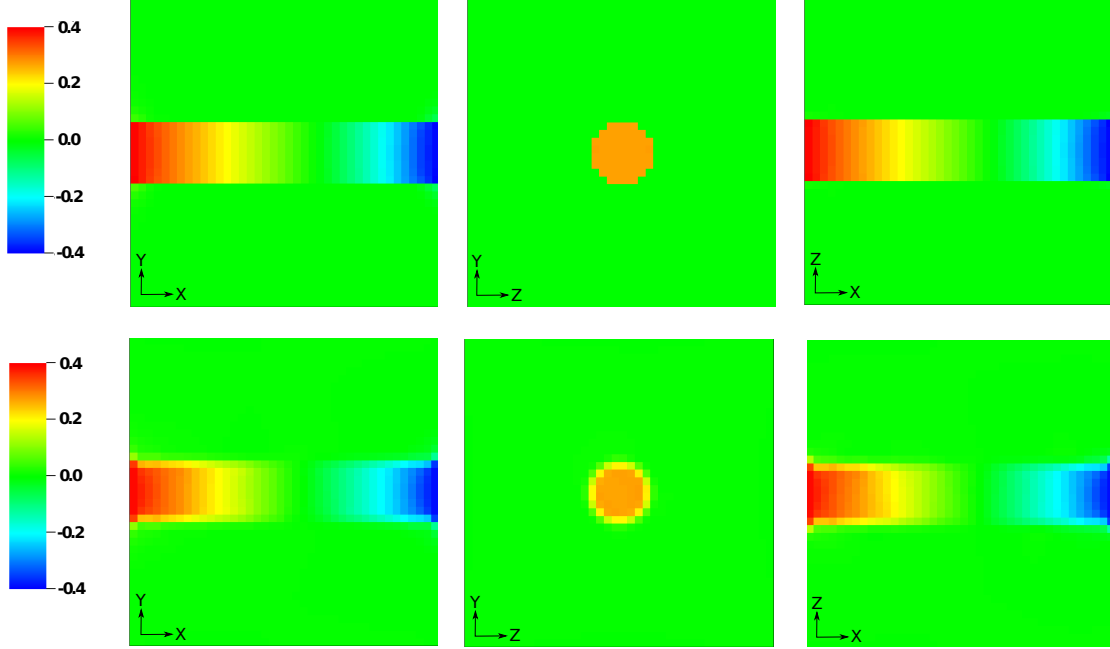


Figure 12: Comparison of the steady-state pressure field along the planes $z = 2.5$, $y = 2.5$, and $x = 1$ obtained using the present IIM (top panel) and the standard IB method (bottom panel) with a relatively coarse locally refined grid comprising $N = 3$ levels. With the IB method, spurious pressure oscillations are clearly observed, whereas the interface method yields a sharp pressure profile throughout the computational domain.

in which $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ and R is the radius of the pipe. The physical domain is $\Omega = [0, L]^3$. A horizontal pipe of length L extends across the middle of the domain and is described using quadrilateral surface elements with $M_{\text{fac}} = 2$, see Fig. 11(a). A pressure difference of $2p_0$ is applied across the pipe. As in the two-dimensional case, the boundary conditions at the inlet and outlet are imposed through applying a combination of normal-traction and zero-tangential-slip boundary conditions [67]. Solid-wall boundary conditions are imposed along the remaining parts of $\partial\Omega$. The computational domain is discretized using an adaptively refined Cartesian grid. With N grid levels and a refinement ratio of two between levels, the Cartesian grid spacing on the finest grid level is $h_{\text{finest}} = 2^{-(N-1)}h_{\text{coarsest}}$, with $h_{\text{coarsest}} = \frac{L}{8}$ being the grid spacing on the coarsest level. The time step size is $\Delta t = 0.05h_{\text{finest}}$, giving a maximum advective CFL number of approximately 0.05–0.1, and the spring stiffness is set to $\kappa = 2.2 \times 10^{-3}/\Delta t^2$. Other material and simulation parameters are $L = 5$, $R = 0.5$, $\rho = 1$, $\mu = 0.01$, and $p_0 = 0.4$, resulting in a Reynolds number of $\text{Re} = 100$.

As in the plane flow case, we also consider an inclined configuration. In the skew case, the structure is placed in the same cubic Cartesian domain but is rotated around the z -axis with an angle of $\theta = \pi/12$; see Fig. 11(b). At the inlet and outlet, we specify normal traction boundary conditions and tangential velocity boundary conditions that are consistent with the exact solution, and solid wall boundary conditions are imposed along the remainder of $\partial\Omega$. Other simulation parameters are the same as the grid-aligned case.

To illustrate the improvement of the present IIM over the conventional IB method, Fig. 12 shows the pressure fields obtained using both methods on a relatively coarse locally refined grid with $N = 3$ levels, which yields an effective grid spacing of $h = 0.156$. Cross sectional views of the planes $z = 2.5$, $y = 2.5$, and $x = 1$ are shown. The IIM clearly yields a more sharply resolved pressure field. The solution accuracy of the velocity using the present immersed interface approach is qualitatively evaluated in Fig. 13. As in the two-dimensional case, the IIM sharply resolves the velocity field. This figure also examines the maximum structural displacement magnitude on the surface of the cylinder. The maximum value is on the order of 3.3×10^{-3} , which is approximately 1/50 of the Cartesian grid cell width h_{finest} .

Finally, a convergence study is performed by systematically increasing the number N of AMR levels.

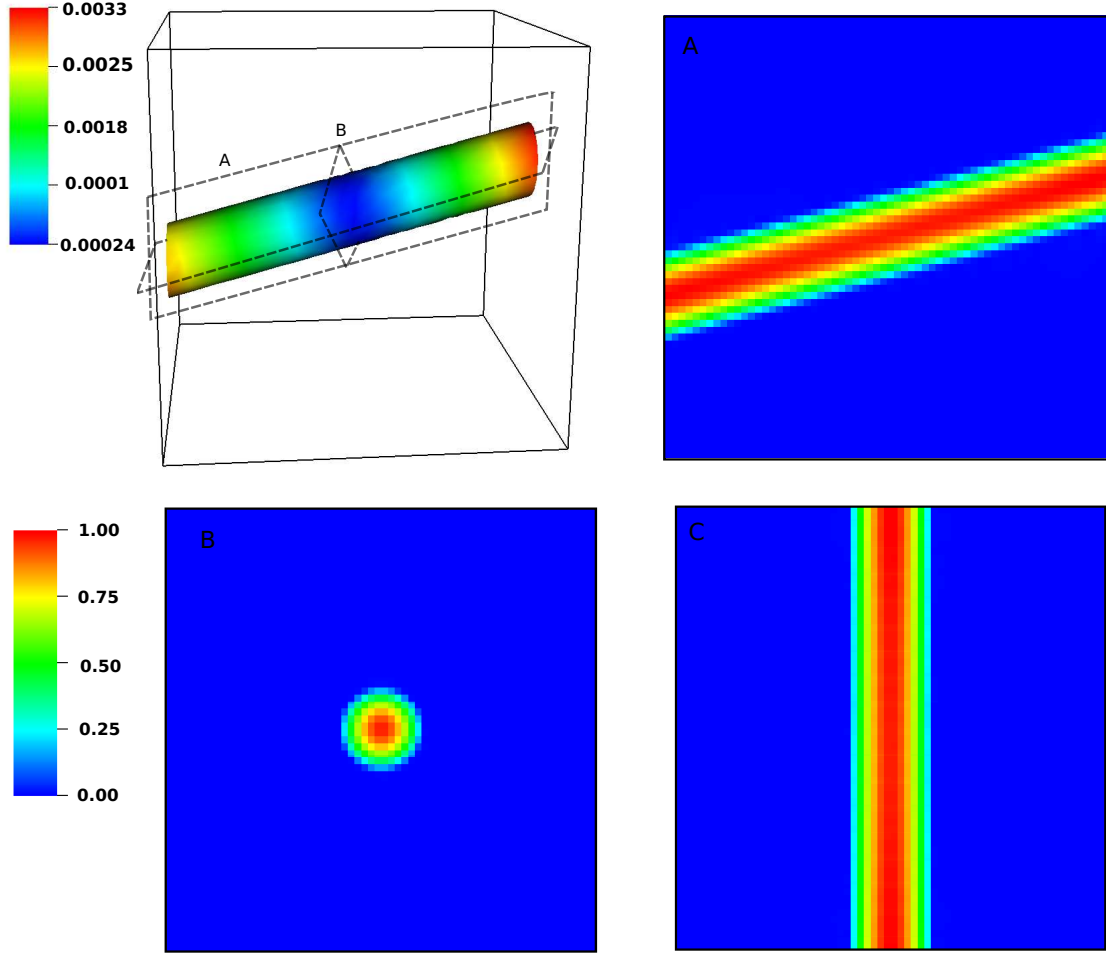


Figure 13: Maximum structural displacement magnitude for the three-dimensional slanted channel (top left panel). Section views of velocity magnitudes at different planes are shown in panels A, B, and C. In this computation, the effective fine grid resolution is $h_{\text{finest}} = 0.156$, and the maximum structural displacement error is approximately 3.3×10^{-3} , which is approximately a factor of 50 *smaller* than h_{finest} .

Fig. 14 reports the L^2 and L^∞ errors in Eulerian velocity and pressure. The method yields second-order convergence rates for the velocity in the L^2 norm for both grid-aligned and skew configurations. Slightly less than second-order accuracy is observed in the L^2 norm of the pressure, and although less than first-order convergence is observed for p in the L^∞ norm over the entire domain, first-order pointwise convergence is recovered in the restricted domain Ω^* . Fig. 15 summarizes the convergence results on the Lagrangian mesh. Second-order convergence rates are achieved for the displacement in both L^2 and L^∞ error norms. Second-order convergence is also observed for the Lagrangian displacement and velocity in the L^2 norm, whereas between first- and second-order convergence is observed for the velocity in the L^∞ norm. The Lagrangian pressure converges at approximately second order in the L^2 norm and between first- and second-order in the L^∞ norm. First-order convergence is observed for the wall shear stress in both the L^2 and L^∞ norms. See A for tabulated errors of this example.

4.2 Circular Couette flow in two and three dimensions

This section considers the steady shear-driven Couette flow of a fluid confined between two rotating circular cylinders in two and three spatial dimensions. This problem is an excellent example to test the tangential portion of the interfacial force on a curved geometry. Fig. 16 provides a schematic of the geometry and

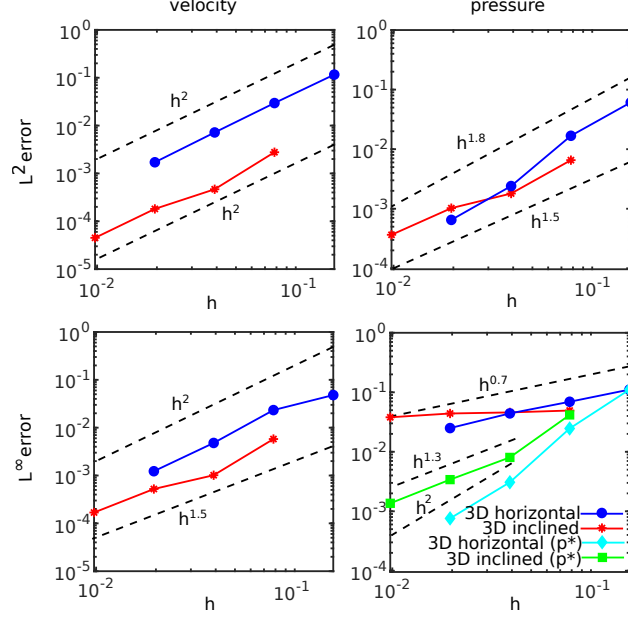


Figure 14: Convergence in the L^2 and L^∞ norms of the error in the Eulerian velocity and pressure for the horizontal and inclined three-dimensional Hagen-Poiseuille flow. Simulation parameters include $\text{Re} = 100$, $\Delta t = 0.1h_{\text{finest}}$, and $M_{\text{fac}} = 2$.

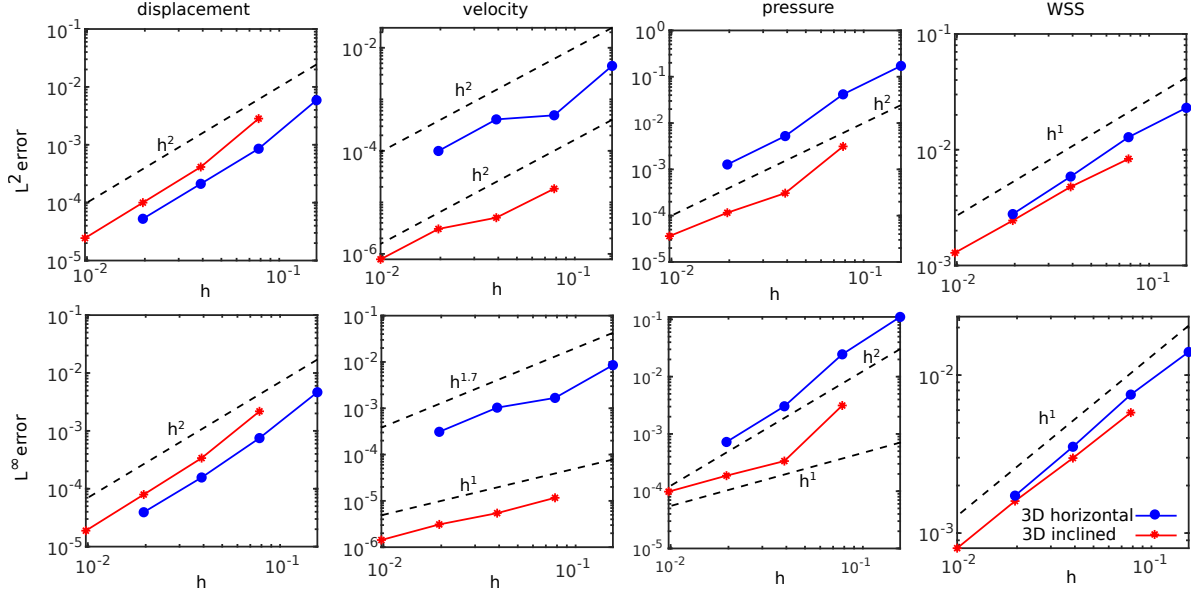


Figure 15: Convergence in the L^2 and L^∞ norms of the error in the Lagrangian displacement, velocity, pressure, and wall shear stress (WSS) for the horizontal and inclined three-dimensional Hagen-Poiseuille flow. Simulation parameters include $\text{Re} = 100$, $\Delta t = 0.1h_{\text{finest}}$, and $M_{\text{fac}} = 2$.

dimensions of the circular Couette flow in two dimensions Analytical solutions for the steady-state velocity and the pressure for the two cylinders co-centered at (x_0, y_0) are

$$u(x, y) = \begin{cases} -\omega_1(y - y_0), & 0 \leq r \leq R_1, \\ -(y - y_0)(A + \frac{B}{r^2}), & R_1 < r < R_2, \end{cases} \quad (46)$$

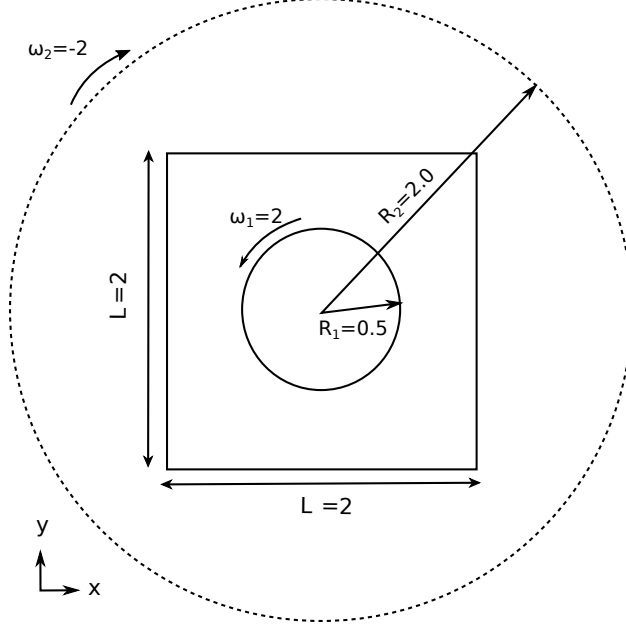


Figure 16: Geometry and dimensions of the circular Couette flow.

$$v(x, y) = \begin{cases} \omega_1(x - x_0), & 0 \leq r \leq R_1, \\ (x - x_0)(A + \frac{B}{r^2}), & R_1 < r < R_2, \end{cases} \quad (47)$$

and

$$p(x, y) = \begin{cases} \frac{\omega_1^2 r^2}{2} + p_0, & 0 \leq r \leq R_1, \\ \frac{A^2 r^2}{2} - \frac{B^2}{2r^2} + AB \cdot \ln r^2, & R_1 < r < R_2, \end{cases} \quad (48)$$

in which $A = \frac{\omega_2 R_2^2 - \omega_1 R_1^2}{R_2^2 - R_1^2}$, $B = \frac{(\omega_1 - \omega_2) R_1^2 R_2^2}{R_2^2 - R_1^2}$, $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$, and p_0 is an arbitrary constant value. We choose $R_1 = 0.5$, $R_2 = 2.0$, $\omega_1 = 2$, and $\omega_2 = -2$. The computational domain is $\Omega = [-1, 1]^2$, and $(x_0, y_0) = (0, 0)$ so that only the inner cylinder is embedded in the physical domain. Dirichlet conditions for the velocity are imposed along $\partial\Omega$ using the analytic relation. The time step size is set to $\Delta t = 0.05h$, yielding a CFL number between 0.05 and 0.1. The penalty parameters are chosen to be $\kappa = 7 \times 10^{-3}/\Delta t^2$ and $\eta = 0$. Other simulation parameters are $M_{\text{fac}} = 2$, $\rho = 1$, and $\mu = 0.01$. The Reynolds number, based on the velocity and diameter of the inner circle, is $\text{Re} = \frac{2\rho R_1^2 \omega_1}{\mu} = 100$.

A three-dimensional extension of the problem is also considered. The same simulation parameters are used as in the two-dimensional case, except that here, the domain is $\Omega = [-1, 1] \times [-1, 1] \times [-2, 2]$. Periodic boundary conditions are applied in the z -direction. The computational domain is discretized using an N -level locally refined Cartesian grid with $h = h_{\text{finest}} = 2^{-(N-1)}h_{\text{coarsest}}$ and $h_{\text{coarsest}} = \frac{L}{8}$ being the coarsest level. Because Re is relatively low, we do not observe Taylor vortices, but the additional dimension helps to further verify the implementation and consistency of results in two and three spatial dimensions.

Fig. 17 compares the performance of the present IIM and the IB method for the two-dimensional steady-state velocity and pressure profiles of the present approach along $x = 0$ using grid spacing $h = 0.0625$. The agreement between the analytic solution and the results generated by the IIM is outstanding.

We also perform a convergence study for both the two- and three-dimensional cases. As seen in Fig. 18 the method successfully achieves global second-order convergence rate of the Eulerian velocity in both two and three dimensions while yielding slightly less than second-order accuracy in the L^∞ norm. At least 1.5-order global convergence rates are obtained for the Eulerian pressure along with first-order convergence rates for the Eulerian pressure on Ω^* . The method also achieves pointwise second-order accuracy in the Lagrangian displacement and velocity along with between first- and second-order pointwise convergence in

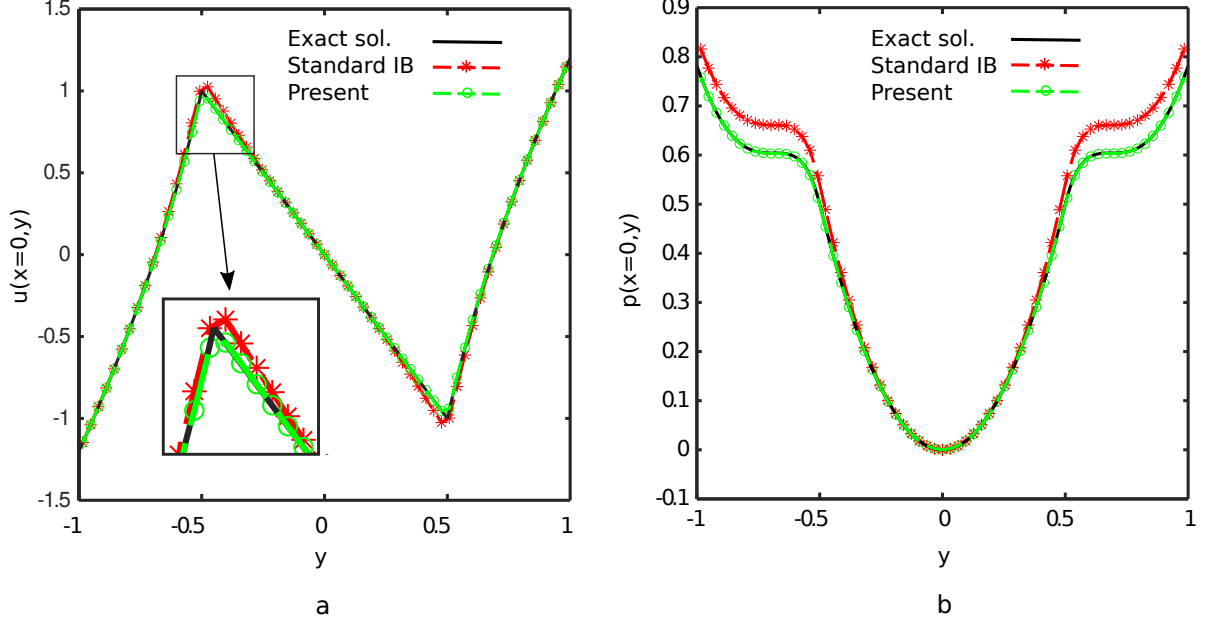


Figure 17: (a) u -velocity and (b) pressure profiles for the IIM and IB solutions to the two-dimensional Couette flow at $Re = 100$ and $h = 0.0625$. The present IIM is in outstanding agreement with the analytic solution.

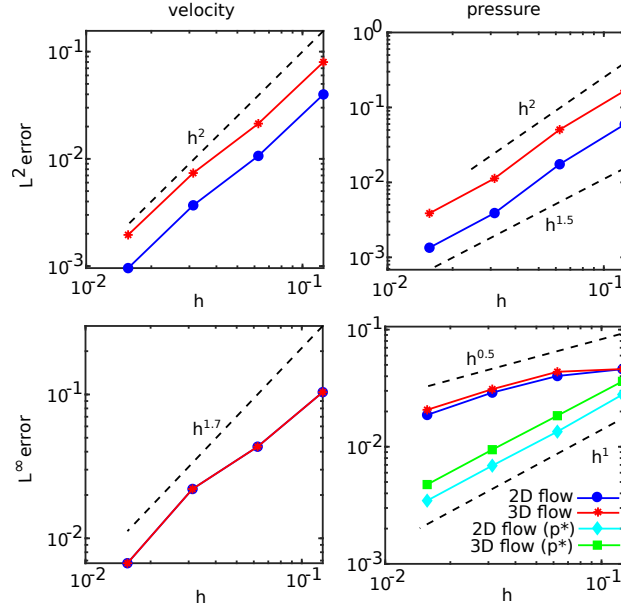


Figure 18: Convergence in the L^2 and L^∞ norms of the error in the Eulerian velocity and pressure for the two- and three-dimensional circular Couette flows. Simulation parameters include $Re = 100$, $\Delta t = 0.05 h_{\text{finest}}$, and $M_{\text{fac}} = 2$.

the Lagrangian pressure and wall shear stress as seen in Fig. 19. Note that the Lagrangian pressure and wall shear stress are computed from the side between the two cylinders.

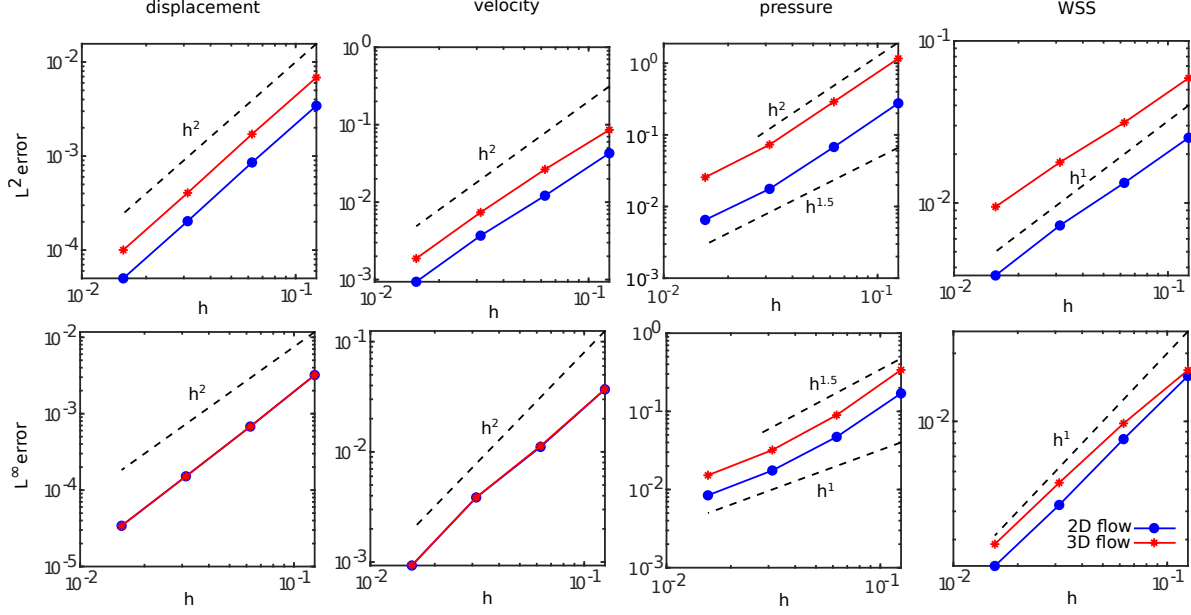


Figure 19: Convergence in the L^2 and L^∞ norms of the Lagrangian displacement, velocity, pressure, and wall shear stress (WSS) for the two- and three-dimensional circular Couette flows. The Lagrangian pressure and wall shear stress are computed from the side between the two cylinders. Simulation parameters include $Re = 100$, $\Delta t = 0.05h_{\text{finest}}$, and $M_{\text{fac}} = 2$.

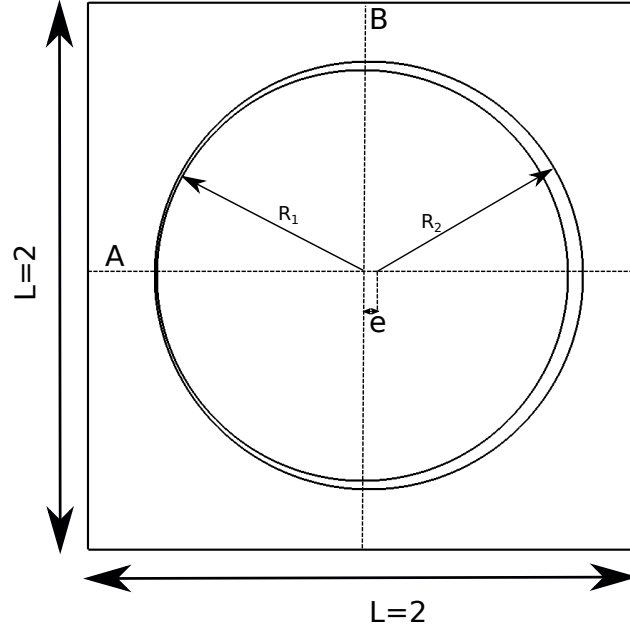


Figure 20: Geometry and dimensions of the eccentric rotating cylinder test. The inner cylinder is centered at the origin, and the outer one is centered at $(e, 0) = (3/128, 0)$.

4.3 Flow within eccentric rotating cylinders

This section considers the two-dimensional flow in the small gap between two eccentric rotating cylinders. This example yields nontrivial fluid dynamics and includes immersed interfaces that are nearly in contact. Fig. 20 provides a schematic of the problem geometry and dimensions. The parameter set including the

dimensions of the problem are the same as Fai and Rycroft [85]. The computational domain is taken to be $\Omega = [-1, 1]^2$, and $R_1 = 3/4$ and $R_2 = 3/4 \times (1 + 1/24)$ are the radii of inner and outer cylinders, respectively. The inner cylinder is centered at the origin, and the outer cylinder is centered at $(e, 0) = (3/128, 0)$. The outer cylinder is set to be stationary while the inner cylinder rotates angular velocity $\omega_1 = 8.33 \times 10^{-4}$. Asymptotic analytical solutions can be derived for the velocity and pressure of the flow within long eccentric cylinders using lubrication theory [86],

$$u(x, y) = \begin{cases} -\omega_1 y, & \sqrt{x^2 + y^2} < R_1, \\ -\omega_1 y \left(1 - \gamma - \frac{3\epsilon(\gamma - \gamma^2)(2x + 3\epsilon\sqrt{x^2 + y^2} + \epsilon^2 x)}{(2 + \epsilon^2)(1 + \epsilon x)}\right), & \sqrt{x^2 + y^2} \geq R_1 \text{ and } \sqrt{(x - e)^2 + y^2} \leq R_2, \\ 0, & \sqrt{(x - e)^2 + y^2} > R_2, \end{cases} \quad (49)$$

$$v(x, y) = \begin{cases} \omega_1 x, & \sqrt{x^2 + y^2} < R_1, \\ \omega_1 x \left(1 - \gamma - \frac{3\epsilon(\gamma - \gamma^2)(2x + 3\epsilon\sqrt{x^2 + y^2} + \epsilon^2 x)}{(2 + \epsilon^2)(\sqrt{x^2 + y^2} + \epsilon x)}\right), & \sqrt{x^2 + y^2} \geq R_1 \text{ and } \sqrt{(x - e)^2 + y^2} \leq R_2, \\ 0, & \sqrt{(x - e)^2 + y^2} > R_2, \end{cases} \quad (50)$$

$$(51)$$

and

$$p(x, y) = \begin{cases} \frac{\omega_1^2(x^2 + y^2)}{2} + p_0, & \sqrt{x^2 + y^2} < R_1, \\ \left(\frac{6\epsilon\mu R_1}{c^2}\right) \left(\frac{2y(x^2 + y^2) + \epsilon xy}{(2 + \epsilon^2)(x^2 + y^2 + \epsilon x)}\right), & \sqrt{x^2 + y^2} \geq R_1 \text{ and } \sqrt{(x - e)^2 + y^2} \leq R_2, \\ 0, & \sqrt{(x - e)^2 + y^2} > R_2, \end{cases} \quad (52)$$

in which the nondimensional thickness is $\epsilon = (R_2 - R_1)/R_1$, the radial clearance is $c = R_2 - R_1$, and p_0 is an arbitrary constant. The reduced radial coordinate parameter is $\gamma = (x^2 + y^2 - R_1\sqrt{x^2 + y^2})/(R_2 - R_1 + \epsilon x)$. A combination of zero normal-traction and zero tangential velocity boundary conditions is imposed on $\partial\Omega$.

The steady state velocity and pressure solutions of the standard IB method and the present approach are obtained using two different mesh sizes. Velocity and pressure profiles are respectively visualized along sections A and B shown in Fig. 20. As in the previous example, a piecewise linear regularized delta function is used to spread the force in the standard IB method. The penalty parameters are chosen to be $\kappa = 2.0 \times 10^{-4}/\Delta t^2$ and $\eta = 0$. In the first case, a relatively coarse 128×128 Cartesian grid is used, resulting in the two cylinders coming as close as $h/2$. Fig. 21 compares the computed results to the asymptotic solution using this coarse Cartesian grid resolution. Although the gap is clearly underresolved in this case, the IIM gives much better accuracy in the velocity field than the standard IB method, which yields large overshoots near the boundaries. We also use a relatively fine 512×512 Cartesian grid is used, resulting in a minimal gap of size $2h$ between the cylinders. Fig. 22 shows that the IIM yields accurate approximations of both the velocity and pressure in this case.

4.4 Flow past a stationary cylinder

Flow past a stationary cylinder is a widely used benchmark problem for testing numerical methods involving immersed boundaries. Here, the immersed boundary is a disc centered at the origin with diameter $D = 1$. The physical domain is $\Omega = [-15, 45] \times [-30, 30]$, a square of length $L = 60$. This domain configuration corresponds to the problem setup in Taira and Colonius [87]. A uniform inflow velocity $U = (1, 0)$ is imposed on the left boundary ($x = -15$), and zero normal traction and tangential velocity is imposed at the right boundary ($x = 45$) as an outflow condition. Along the bottom ($x = -30$) and top ($x = 30$) boundaries, the normal velocity and tangential traction are set to zero.

We set $\rho = 1$ and use the inflow velocity U as the characteristic velocity. The Reynolds number is $\text{Re} = \frac{\rho U D}{\mu}$. Reynolds numbers between 20 to 200 are considered. The computational domain is discretized using a locally refined grid with $N = 6$ nested grid levels and a refinement ratio of two between levels. The Cartesian grid spacing on the coarsest level is $h_{\text{coarsest}} = \frac{L}{64}$, and $h_{\text{finest}} = 2^{-(N-1)}h_{\text{coarsest}}$ is the grid spacing on the finest grid level. The time step size is $\Delta t = 0.05h_{\text{finest}}$, yielding a maximum advective CFL number

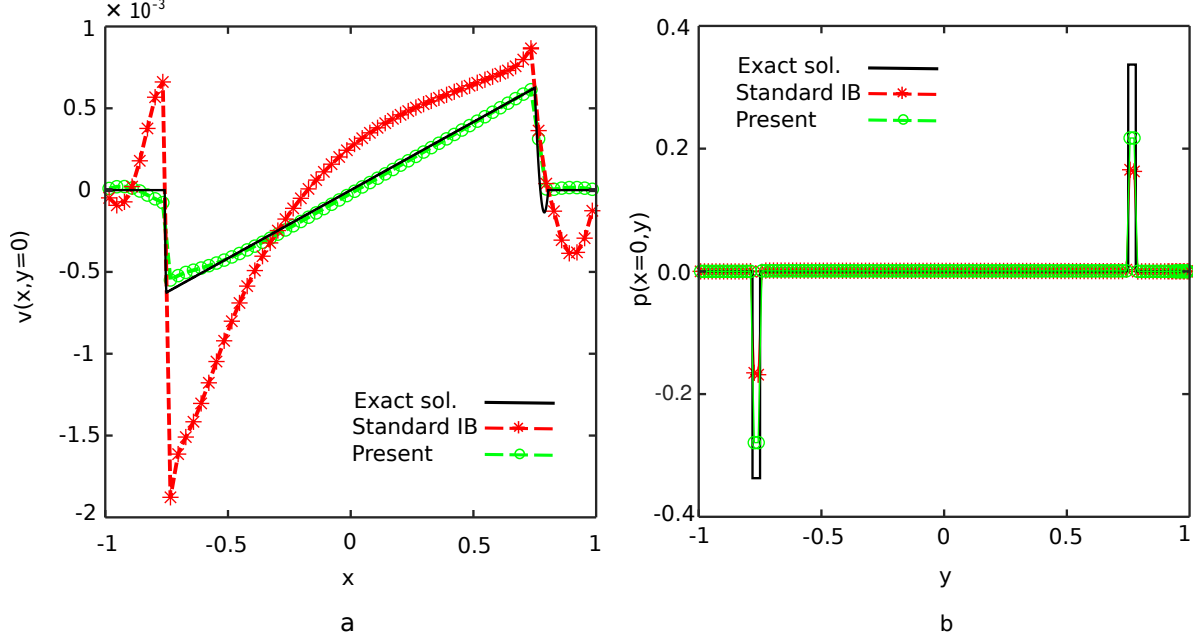


Figure 21: (a) v -velocity and (b) pressure profiles for the IIM and IB solutions of the two-dimensional flow within the eccentric rotating cylinders on a relatively coarse 128×128 Cartesian grid, for which the minimum distance between the cylinders is $h/2$. The present IIM yields results that are in reasonable quantitative agreement with the asymptotic solution in this underresolved case.

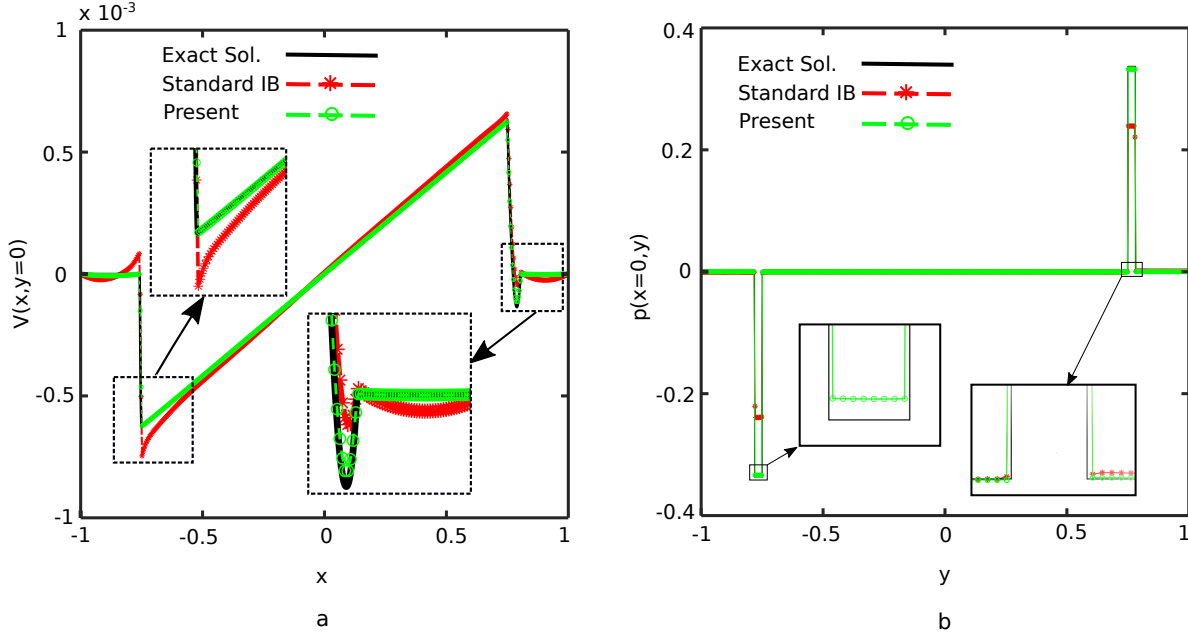


Figure 22: (a) v -velocity and (b) pressure profiles for the IIM and IB solutions of the two-dimensional flow within eccentric rotating cylinders on a relatively fine 512×512 Cartesian grid, for which the minimum distance between the cylinders is $2h$. The present IIM yields results that are in excellent quantitative agreement with the asymptotic solution even in this marginally resolved case.

in the range 0.1–0.2. Values of κ and η are tuned to ensure both rigidity of the structure and stability of the dynamics.

To quantitatively assess the computed dynamics, we compute nondimensional quantities including the drag coefficient C_D and lift coefficient C_L as,

$$(C_D, C_L) = \frac{-\int_{\Gamma_0} \mathbf{F}(\mathbf{X}, t) \, dA}{\frac{1}{2}\rho U^2 D}, \quad (53)$$

and the Strouhal number,

$$\text{St} = \frac{f_s D}{U}, \quad (54)$$

in which F^x and F^y are the x and y components of the penalty force, and f_s is the vortex shedding frequency. Table 1 lists the drag coefficient (C_D), recirculation length (L_{wake}), and angle of separation (θ_s) for $\text{Re} = 20$ and $\text{Re} = 40$. For this range of Reynolds numbers, the flow separates from the back of the cylinder, and a pair of vortices form that gradually approach a steady state. Comparisons of the results obtained by our method with previous numerical and experimental results show excellent quantitative agreement in all flow characteristics.

A transition from steady flow to alternate vortex shedding occurs for $40 < \text{Re} < 50$. The present IIM recovers this transition, as demonstrated in Fig. 23. Although an initial instability decays over time at $\text{Re} = 40$, the same instability at $\text{Re} = 50$ eventually leads to the well-known von Karman vortex street. Vortex shedding continues for $\text{Re} = 100$ and $\text{Re} = 200$, and the flow becomes increasingly unsteady. Fig. 24 details the lift and drag coefficients at $\text{Re} = 100$.

Fig. 25 shows representative interfacial displacement and velocity magnitudes at $\text{Re} = 200$. The maximum displacement magnitude is approximately 2.6% that of the background grid spacing ($h_{\text{finest}} = \frac{60}{64 \times 2^5} = 0.0293$), and the velocity magnitude is approximately 0.016% that of the free stream flow velocity ($U = 1$).

Fig. 26 compares the vorticity fields generated by the immersed interface and IB methods at $\text{Re} = 200$. It is clear that the IIM gives a very sharply resolved vorticity profile while allowing for essentially no flow in the interior of the structure. This is in clear contrast to the conventional IB method, which generates spurious flow regions inside the cylinder, as seen in Fig. 26(b).

Table 2 compares C_L , C_D , and St with previous experimental and computational studies at $\text{Re} = 100$ and $\text{Re} = 200$. The force coefficients fall within the range of values reported in previous studies.

We also investigate the effects of varying the relative mesh spacing parameter M_{fac} . A range of relative mesh spacing values are considered. Fig. 27 shows the lift and drag coefficients at $\text{Re} = 200$ with $M_{\text{fac}} = 1.5, 2, 3$, and 4. The results are consistent across a relatively large range of M_{fac} values. This mesh insensitivity is important when modeling complex geometries, for which large variations in the element sizes are likely. We remark that it is not possible to use arbitrarily small values of M_{fac} with the present method. To see why this is, recall that forces are transmitted from the interface to the background Cartesian grid through jump conditions that are determined by intersecting the finite difference stencils against the interface mesh. Forces that are associated with elements that are not intersected by any finite difference stencil will have no physical effect on the flow dynamics. This is physically unstable, and if it occurs in a simulation, the

Table 1: Comparison of computational and experimental values for the flow past a stationary cylinder at $\text{Re} = 20$ and $\text{Re} = 40$. Simulation parameters include $h_{\text{finest}} = 0.0293$, $\Delta t = 0.05h_{\text{finest}}$, and $M_{\text{fac}} = 2$.

	Re = 20			Re = 40		
	L_{wake}	θ_s	C_D	L_{wake}	θ_s	C_D
Tritton (experimental) [88]	-	-	2.22	-	-	1.48
Le et al. [50]	0.93	43.9	2.05	2.35	53.8	1.52
Xu and Wang [47]	0.92	44.2	2.23	2.21	53.5	1.66
Calhoun [89]	0.91	45.5	2.19	2.18	54.2	1.62
Present	0.93	44.4	2.10	2.31	54.1	1.58

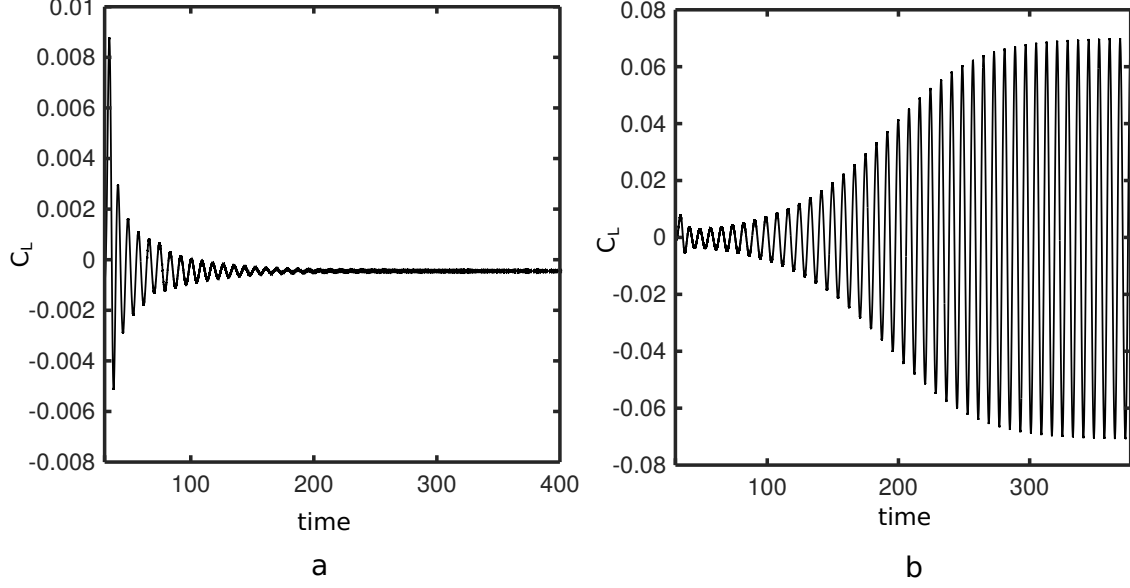


Figure 23: The lift coefficient (C_L) of flow past a stationary cylinder over time for (a) $Re = 40$ and (b) $Re = 50$. The onset of the von Karman vortex street is detected at $Re = 50$. Simulation parameters include $h_{\text{finest}} = 0.0293$, $\Delta t = 0.05h_{\text{finest}}$, and $M_{\text{fac}} = 2$.

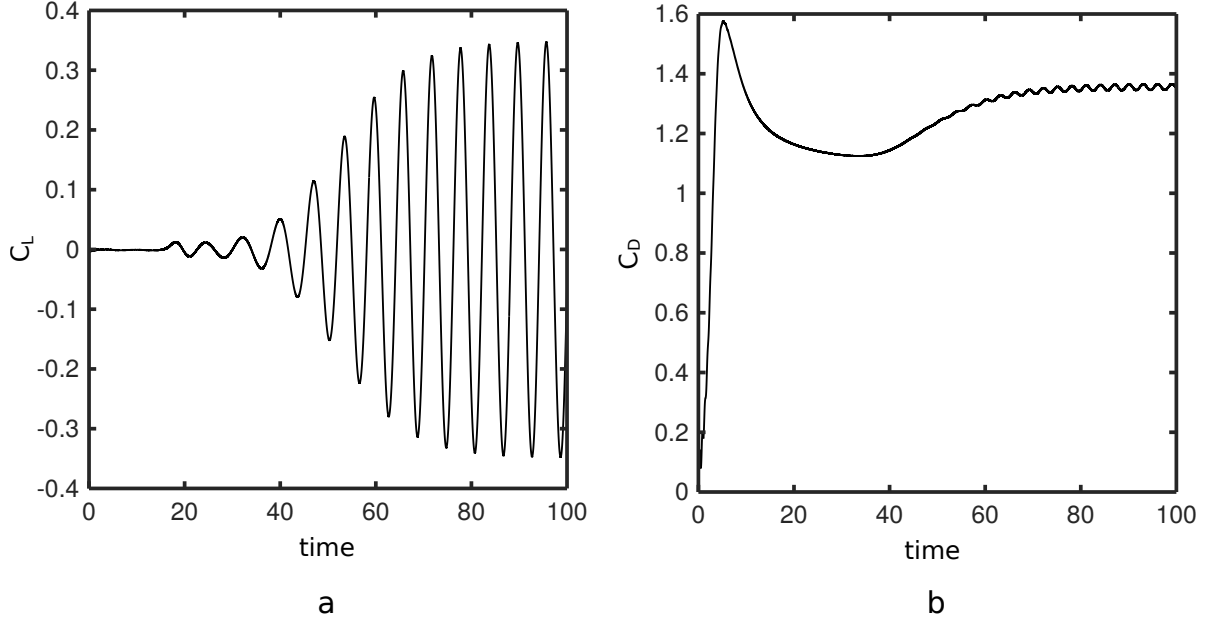


Figure 24: Lift (C_L) and drag (C_D) coefficients over time for flow past a cylinder at $Re = 100$. Simulation parameters include $h_{\text{finest}} = 0.0293$, $\Delta t = 0.05h_{\text{finest}}$, and $M_{\text{fac}} = 2$.

computation also will generally become unstable. Consequently, we consider values of M_{fac} that are large enough to ensure that all surface elements are pierced by at least one finite difference stencil.

4.5 Flow past a spinning cylinder

We also consider the performance of the method in determining the lift coefficient for flow over a spinning cylinder. The computational domain and the Eulerian boundary conditions are the same as the case of the

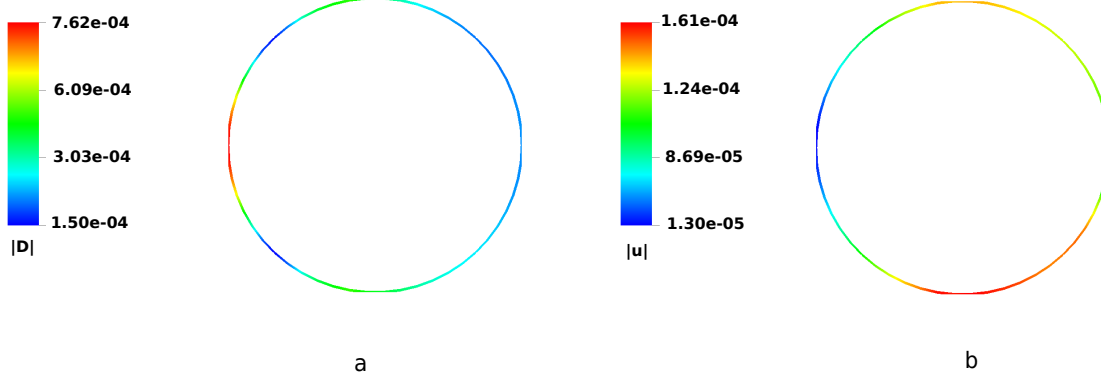


Figure 25: Interfacial (a) displacement and (b) velocity magnitudes for flow around a stationary cylinder at $Re = 200$ using the present IIM, plotted at the same time as shown in Fig. 26. Simulation parameters include $h_{\text{finest}} = 0.0293$, $\Delta t = 0.05h_{\text{finest}}$, and $M_{\text{fac}} = 2$. The maximum displacement error is approximately 2.6% that of the background grid spacing h_{finest} , and the velocity error is approximately 0.016% that of the free stream flow velocity.

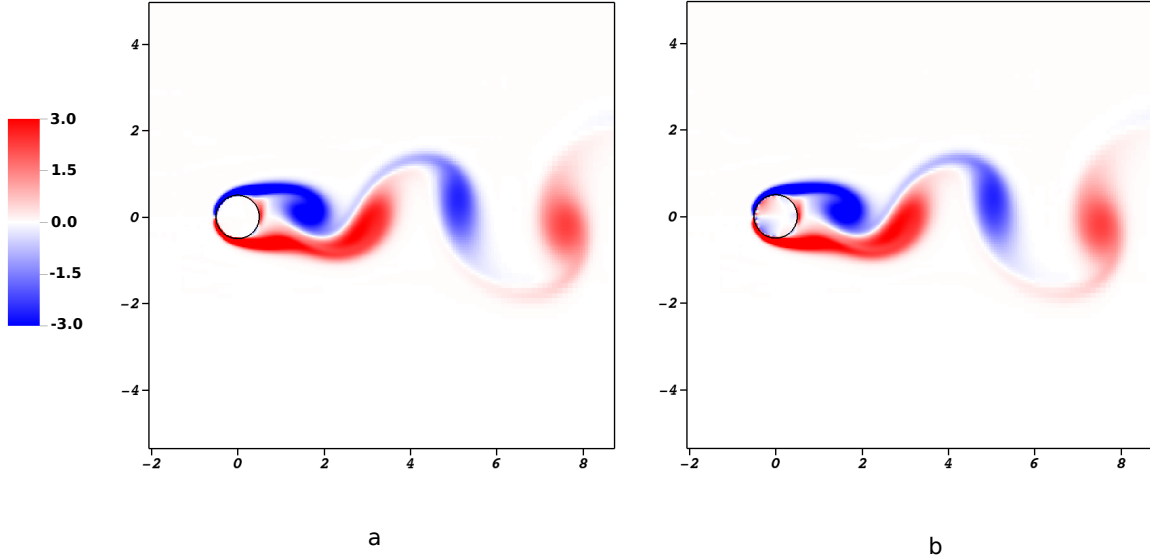


Figure 26: Vorticity fields for flow past a cylinder at $Re = 200$ for (a) the present IIM and (b) the conventional IB method. Notice that the conventional IB method generates spurious interior flows that are eliminated with the present IIM.

stationary cylinder in Sec. 4.4. The cylinder is prescribed to rotate about its central axis at nondimensional rotation rate $\alpha = \frac{\omega R}{U}$, in which ω is the angular velocity. Fig. 28 compares the lift coefficient generated by the present method to previous work [94, 95, 74, 96, 97] for α in the range 0–3 at $Re = 5$ and 20. The C_L values generated by the present IIM are in excellent agreement with prior work for the entire range of α values considered.

Fig. 29 shows the vorticity fields for $\alpha = 1, 2, 2.5, 3.5$, and 4.4 at $Re = 200$. As previously demonstrated by Mittal and Kumar [98], the time history of C_L reveals vortex shedding behavior for $\alpha < 1.9$. At higher rotation rates, the flow is expected to achieve a steady state except for $4.34 < \alpha < 4.70$, for which the flow undergoes a different kind of instability, such that only one-sided vortex shedding occurs. Fig. 29 demonstrates that these flow conditions are accurately predicted using the present IIM.

Table 2: Comparison of computational and experimental values of the drag coefficient (C_D), lift coefficient (C_L), and Strouhal numbers (St) for flow past a cylinder at $Re = 100$ and $Re = 200$. Other simulation parameters include $h_{\text{finest}} = 0.0293$, $\Delta t = 0.05h_{\text{finest}}$, and $M_{\text{fac}} = 2$. All values are computational unless otherwise noted.

	Re = 100			Re = 200		
	C_D	C_L	St	C_D	C_L	St
Braza et al. [90]	1.360 ± 0.015	± 0.250	0.160	1.400 ± 0.050	± 0.75	0.200
Liu et al. [91]	1.350 ± 0.012	± 0.339	0.164	1.310 ± 0.049	± 0.69	0.192
Calhoun [89]	1.330 ± 0.014	± 0.298	0.175	1.170 ± 0.058	± 0.67	0.202
Le et al. [50]	1.370 ± 0.009	± 0.323	0.160	1.340 ± 0.030	± 0.43	0.187
Xu and Wang [47]	1.423 ± 0.013	± 0.340	0.171	1.420 ± 0.040	± 0.66	0.202
Lai and Peskin [4]	-	-	0.165	-	-	0.190
Roshko (experimental) [92]	-	-	0.164	-	-	0.190
Williamson (experimental) [93]	-	-	0.166	-	-	0.197
Griffith and Luo [74]	-	-	-	1.360 ± 0.046	± 0.70	0.195
Present	1.370 ± 0.015	± 0.351	0.168	1.390 ± 0.060	± 0.75	0.198

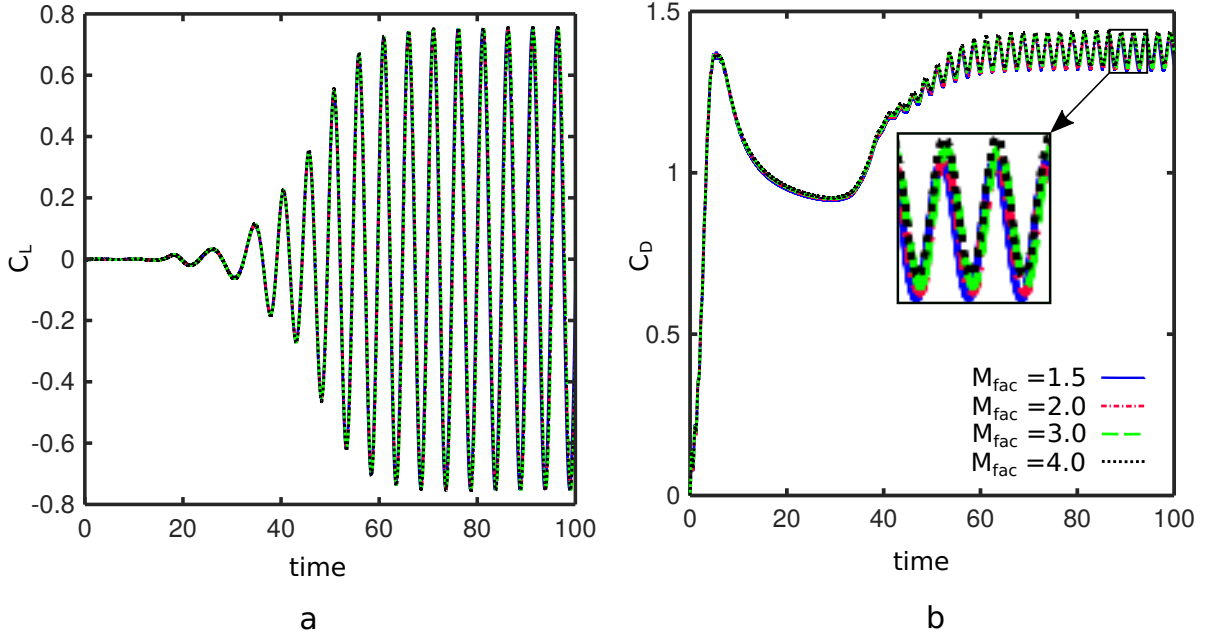


Figure 27: Lift (C_L) and drag (C_D) coefficients for flow past a cylinder at $Re = 200$. The method produces consistent results across a wide range of M_{fac} values.

4.6 Flow past a stationary sphere

This section examines flow past a stationary sphere at low to moderate Reynolds numbers. We consider a sphere with diameter of $D = 1$ centered at the origin and placed in the computational domain $\Omega = [-15, 45] \times [-30, 30] \times [-30, 30]$. A uniform inflow velocity $U = (1, 0, 0)$ is imposed at $x = -15$, and zero normal traction and tangential velocity is imposed at $x = 45$ as an outflow condition. Along the other boundaries, the normal velocity and tangential traction are set to zero.

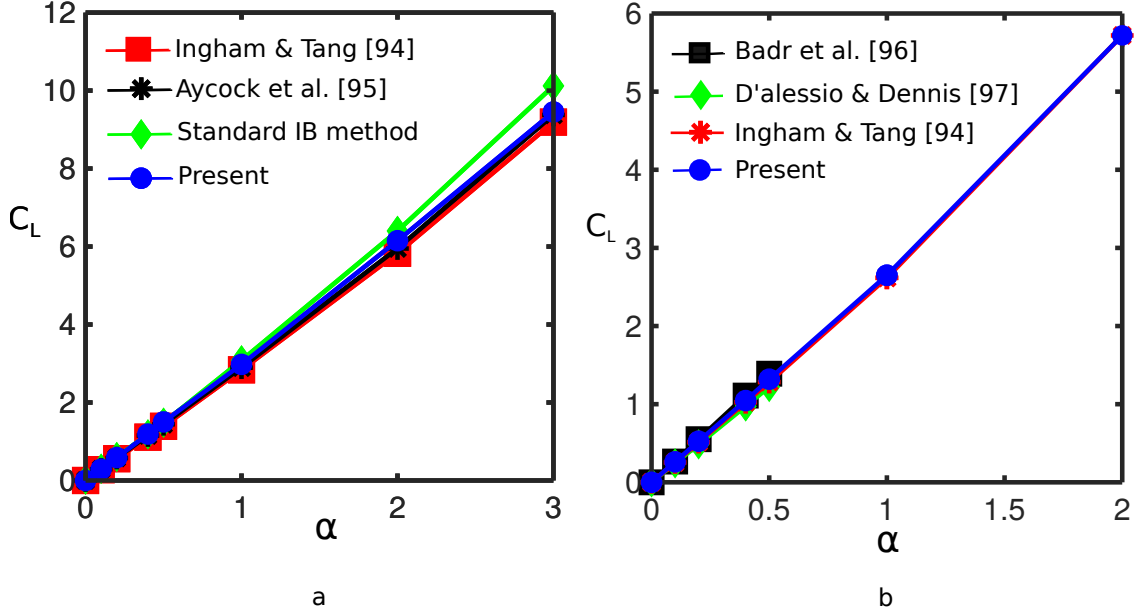


Figure 28: Lift coefficient (C_L) versus nondimensional angular velocity α for the flow past a rotating cylinder at (a) $Re = 5$ and (b) $Re = 20$. Both cases use $M_{fac} = 2$. Results produced by the present method are in excellent agreement with previous numerical studies.

Table 3: Drag coefficients for three-dimensional flow past a sphere at different Reynolds numbers determined by the present IIM and by several prior computational [99, 55, 42] and empirical [100] studies. Simulation parameters include $h_{finest} = 0.0293$, $\Delta t = 0.05h_{finest}$, and $M_{fac} = 2$.

	$Re = 20$	$Re = 100$	$Re = 200$	$Re = 500$
Fornberg [99]	-	1.0852	0.7683	0.4818
Turton and Levenspiel [100]	2.6866	1.0994	0.8025	0.5617
Fadlun et al. [42]	-	1.0794	0.7567	0.4758
Campregher et al. [101]	-	1.1781	0.8150	0.5200
Xu and Wang [55]	2.73	1.15	0.88	-
Present	2.6940	1.0920	0.7852	0.5348

We set $\rho = 1$ and use the inflow velocity U as the characteristic velocity. The Reynolds number is $Re = \frac{\rho U D}{\mu}$, and we consider Reynolds numbers from 20 to 500. The computational domain is discretized using an N -level locally refined grid with a refinement ratio of two between the grid levels. The Cartesian grid spacing on the coarsest level is $h_{coarsest} = \frac{L}{64}$, and $h_{finest} = 2^{-(N-1)}h_{coarsest}$ is the grid spacing on the finest grid level. The time step size is $\Delta t = 0.05h_{finest}$, yielding a maximum CFL number of approximately 0.1–0.2. The surface of the sphere is described using bilinear quadrilateral surface elements with $M_{fac} = 2$. The spring stiffnesses and damping parameter at $Re = 20$ are taken to be $\kappa = 1000$ and $\eta = 5$. The values for $Re = 100$ are $\kappa = 300$ and $\eta = 0.5$. At higher Reynolds numbers of $Re = 200$ and $Re = 500$, only spring forces are used, and the spring stiffness in these cases are chosen to $\kappa = 250$ and $\kappa = 200$, respectively. We compute the force coefficients as

$$(C_D, C_L^y, C_L^z) = \frac{-2 \int_{\Gamma_0} \mathbf{F} \, dA}{A_{proj}}, \quad (55)$$

in which $A_{proj} = \pi/4$ is the projected area of the sphere with $D = 1$. Table 3 compares results generated by the present method to previous computational and experimental studies [99, 100, 42, 101, 55] for Re between

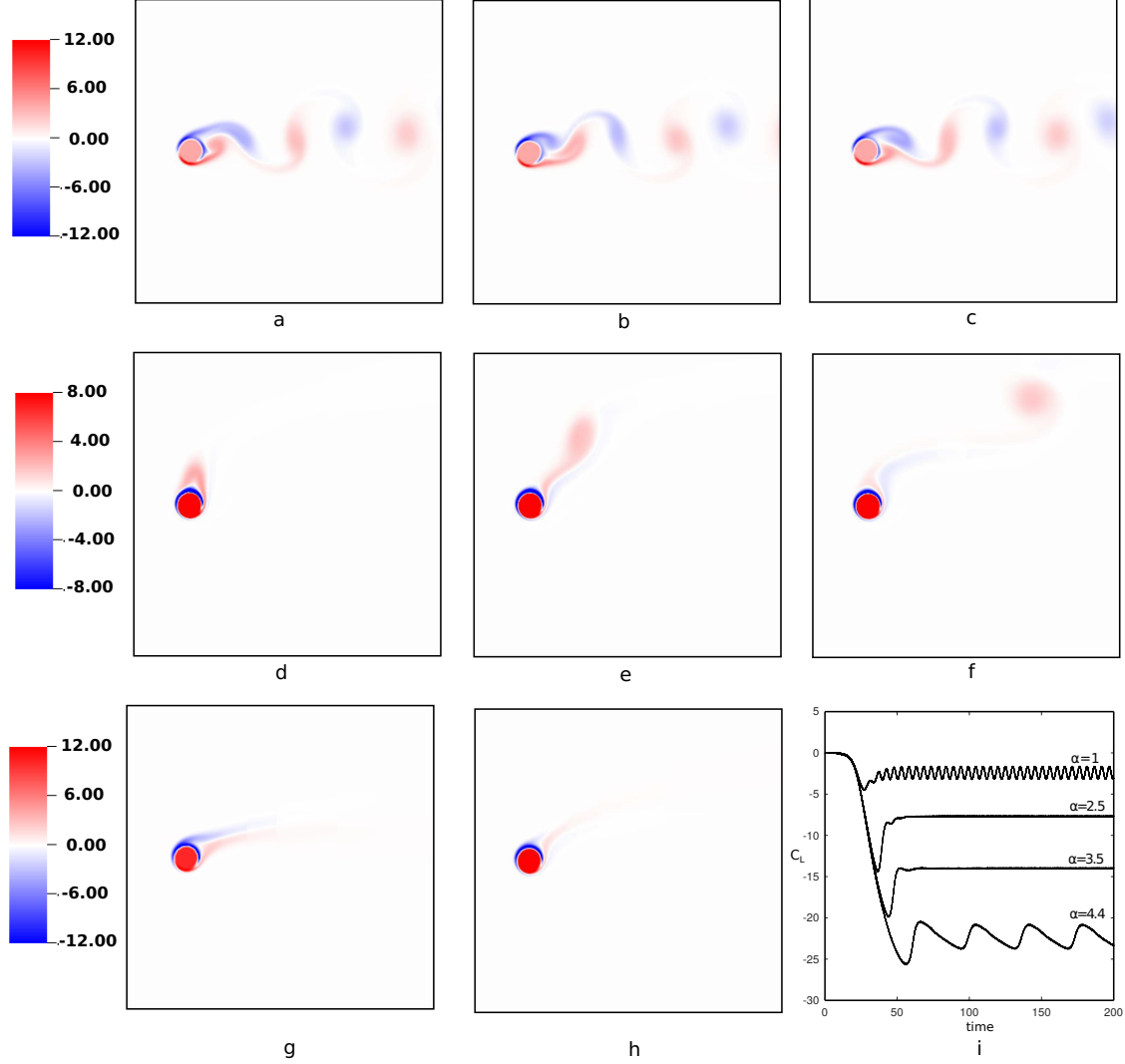


Figure 29: Vorticity fields of flow past a rotating cylinder for four rotation rates at $Re = 200$. Panels (a), (b), and (c) show vorticity fields corresponding to $\alpha = 1.0$ at $t = 128, 130,$ and 132 , respectively. Vorticity contours of $\alpha = 4.4$ are shown in panels (d), (e), and (f) at $t = 132, 138,$ and 144 . Periodic flows with distinct behaviors are observed for $\alpha = 1.0$ and 4.4 , whereas the solutions at $\alpha = 2.5$ shown in panel (g) and at $\alpha = 3.5$ in panel (h) reach steady state. Subfigure (i) shows time histories of the corresponding lift coefficient C_L in the lower bottom panel.

20 to 500. We observe excellent agreement between the values produced by the present method and these previous works.

To test the ability of the method in predicting the dynamics of the flow around the sphere in the transitional regime, simulations are performed at $Re = 200, 260,$ and 300 . It is well established that the onset of the unsteady vortex shedding regime occurs for $290 < Re < 400$ [103, 104]. Fig. 30 shows that unsteady vortex shedding occurs using the present approach at $Re = 300$. Fig. 30 also shows the magnitude of the wall shear stress on the surface of the sphere.

4.7 Flow inside the inferior vena cava

The inferior vena cava (IVC) is a large vein through which deoxygenated blood from the lower and middle body returns to the right atrium of the heart. As a demonstration of the capability of the present method

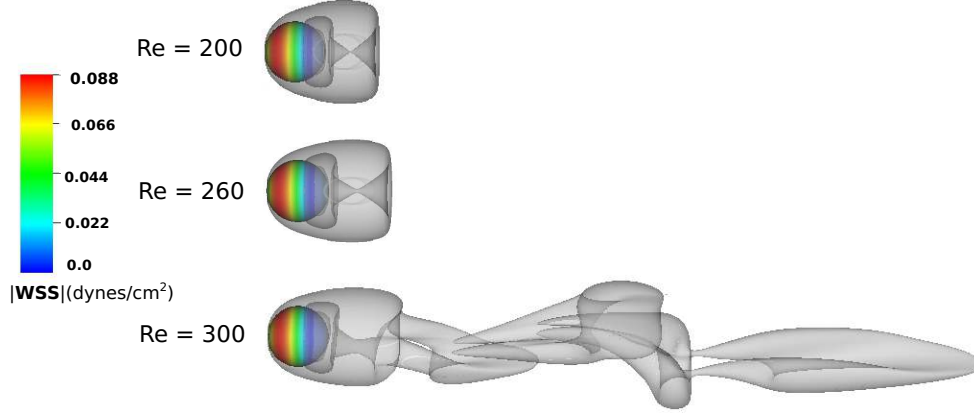


Figure 30: Iso-surface of Q-criterion [102] showing the vortex dynamics for flow past a sphere at $Re = 200$, 260 , and 300 . The color contours on the sphere show the distribution of the wall shear stress, with red and blue being regions of largest and lowest magnitudes, respectively. The onset of the unsteady vortex shedding regime occurs in the range $290 < Re < 400$. The unsteady behavior at $Re = 300$ predicted by the present method agrees with previous studies [103, 104].

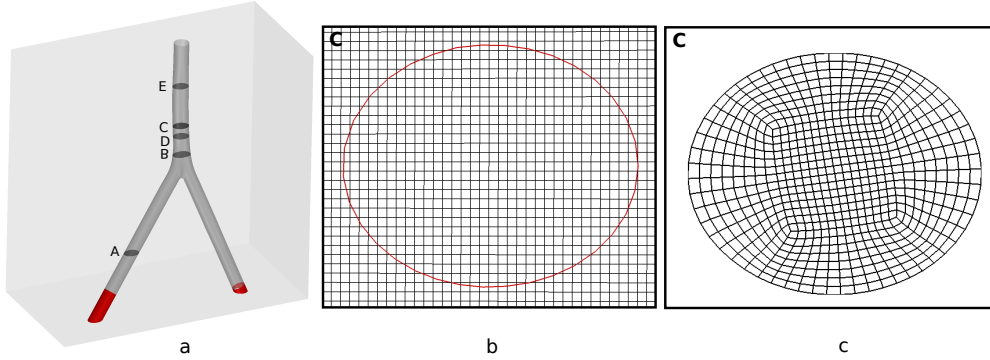


Figure 31: (a) The patient-averaged inferior vena cava (IVC) model geometry used to compare the performance of the present IIM scheme to a body-fitted finite volume simulation performed with OpenFOAM. Comparisons between the two approaches focus on the highlighted sections A–E. (b) The projected Eulerian and interface geometry used in the IIM simulation along section C. (c) The body fitted mesh used in the OpenFOAM simulation along section C. The number of grid cells along the major and minor axes are roughly equal for both simulations.

to handle complex geometries, we consider flow through a model of the human IVC. This geometry used in this section is a modified version of the patient-averaged model by Rahbar et al. [105] that has been recently studied by Craven et al. [106] and Gallagher et al. [107]. Fig. 31(a) shows the geometry of the IVC. The morphological features of the IVC including the flow mixing at the junction of the iliac veins, infrarenal curvature, and non-circular vessel cross sections that all contribute to the complex hemodynamics and make this model a suitable demonstrative case to test the robustness of the present algorithm. The infrarenal IVC downstream of the iliac bifurcation has an average hydraulic diameter of $D_h = 2.8$ cm. A high flow rate of $100 \text{ cm}^3/\text{s}$ is used, which corresponds to exercise flow conditions with a maximum Reynolds number of $Re = 1500$. In accord with the experimental observations of Gallagher et al. [107] in the same model and under the same flow conditions, here we assume that the flow is laminar. The density of the fluid is set to

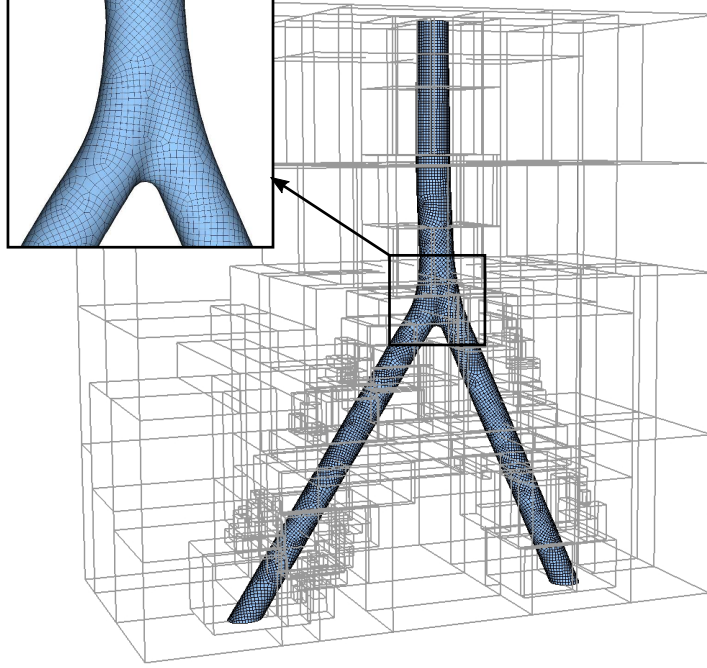


Figure 32: Computational mesh of the IVC including quadrilateral surface elements and a locally refined Cartesian grid.

$\rho = 1.817 \text{ g/cm}^3$, and the viscosity is $\mu = 5.487 \times 10^{-2} \text{ g/cm s}$. For comparison, an additional simulation is performed with the finite volume method using the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) solver in OpenFOAM (version 1812) with second-order accurate spatial discretization schemes. Importantly, the OpenFOAM simulation was performed using a body-fitted block-structured mesh that was generated to approximately match the spatial resolution of the IIM mesh (see Fig. 31 panels b and c) to enable a consistent comparison.

Steady fully-developed parabolic velocity boundary conditions are imposed at the upstream inlets of the iliac veins. The two inlets are circular with a diameter of $D = 2.44 \text{ cm}$, which then transition to an elliptical shape a short distance downstream. Because our current IIM implementation requires applying the Eulerian boundary conditions at the outer faces of the surrounding Cartesian domain, we have slightly extended the IVC geometry at the inlets to be able to apply the corresponding fully developed conditions at the resulting enclosed intersection ellipses such that the flow conditions are the same as in the OpenFOAM simulation; see Fig. 31(a). The surface of the IVC is described using bilinear quadrilateral surface elements with $M_{\text{fac}} \approx 2$, as shown in Fig. 32. The surface mesh of the IVC used in the IIM simulation was constructed using Trellis Mesh Generation Toolkit, which is based on the CUBIT software [108], and the mesh of the OpenFOAM simulation was generated using the Pointwise software. The IVC is embedded in a rectangular computational domain of size $L_x \times L_y \times L_z = 50 \times 25 \times 50 \text{ cm}^3$. This Eulerian domain is discretized using a three-level locally refined grid with a refinement ratio of four between the grid levels, resulting in a grid spacing of $h_{\text{coarsest}} = \frac{25}{16} = 1.5625 \text{ cm}$ on the coarsest level and $h_{\text{finest}} = \frac{25}{16 \times 4^2} \approx 0.098 \text{ cm}$ on the finest grid level. A fixed time step size of $\Delta t = 10^{-4} \text{ sec}$ is used. At the outlet, the normal traction and tangential velocities are set to zero. Solid-wall boundary conditions are imposed along the remainder of $\partial\Omega$.

Fig. 33 shows the Lagrangian displacement and velocity magnitudes of the stationary IVC using the present IIM. The maximum displacement of the mesh stays within 18% of the Eulerian grid spacing, and the maximum velocity is approximately 0.032% that of the mean value velocity at the inlet ($\bar{U} \approx 10.7 \text{ cm/s}$). Numerical experiments show that the fully-developed flows through the iliac veins remain almost completely undisturbed until they reach the confluence of the veins, where the two streams merge and form a high-velocity region in the center of the IVC lumen, with a pair of counter-rotating vortices on either side. This

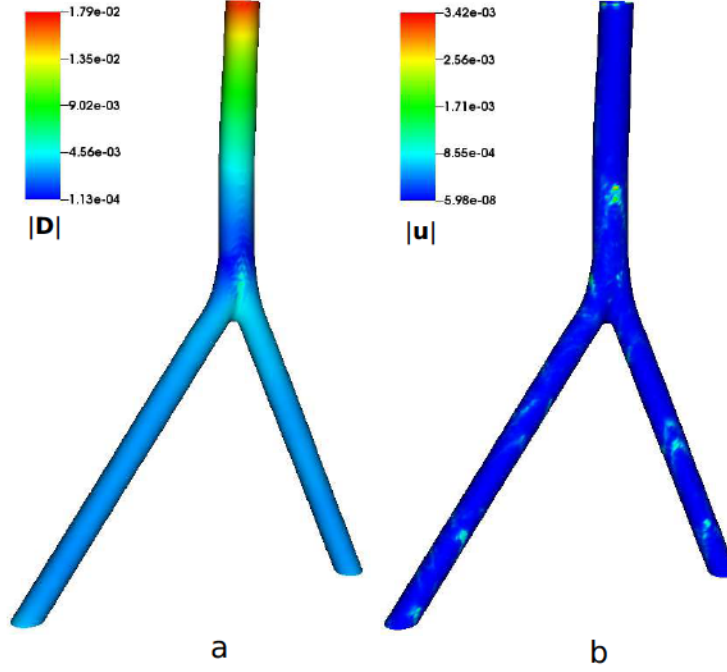


Figure 33: Lagrangian (a) displacement and (b) velocity magnitudes of the rigid IVC mesh using the present IIM. The maximum displacement is approximately 18% that of the background grid spacing, and the maximum velocity value is approximately 0.032% that of the mean value velocity at the inlet ($\bar{U} \approx 10.7$ cm/s).

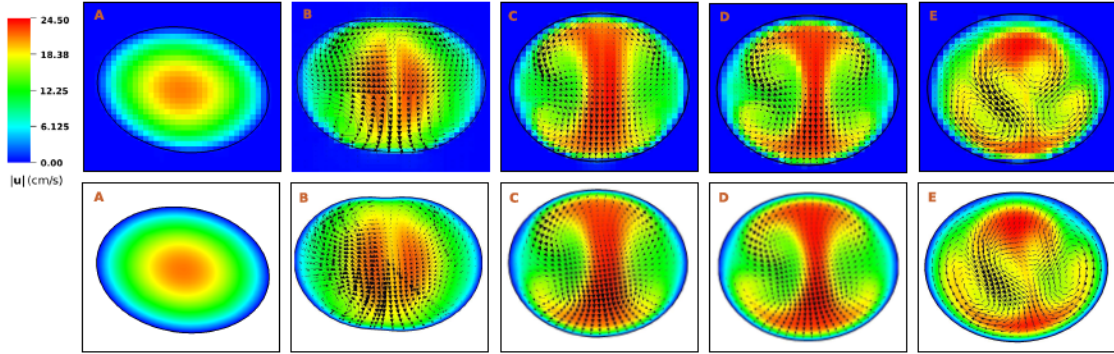


Figure 34: Velocity magnitudes and vorticity vectors for flow in the inferior vena cava (IVC) at exercise flow conditions using the present IIM approach (top panels) and the OpenFOAM simulation (bottom panels). Good agreement between the two solutions is clearly obtained at all cross sections. Note that no additional post-processing smoothing or filtering is performed in the visualization of the results from the present method.

is demonstrated in Fig. 34(a), where velocity contours of both the IIM computation and the OpenFOAM simulation at cross section A are shown. Significant swirl and mixing occur after the confluence of the iliac veins. The velocity contours and the flow patterns are compared between the present method and the OpenFOAM simulation for four slices labeled B, C, D, and E past the confluence region. Fig. 34 shows there is a good agreement in both velocity magnitudes and the flow patterns between the two solutions.

To further quantify and compare the amount of swirl and mixing in the solution, we compute the helicity density $H = \mathbf{u} \cdot \boldsymbol{\omega}$. This parameter is an indicator of local embolus particle transport in cardiovascular flows [109]. Fig. 35 illustrates the contours of helicity for the two solutions using the present IIM approach and OpenFOAM at the critical cross section E previously shown in Fig. 31. Once again, overall there is good

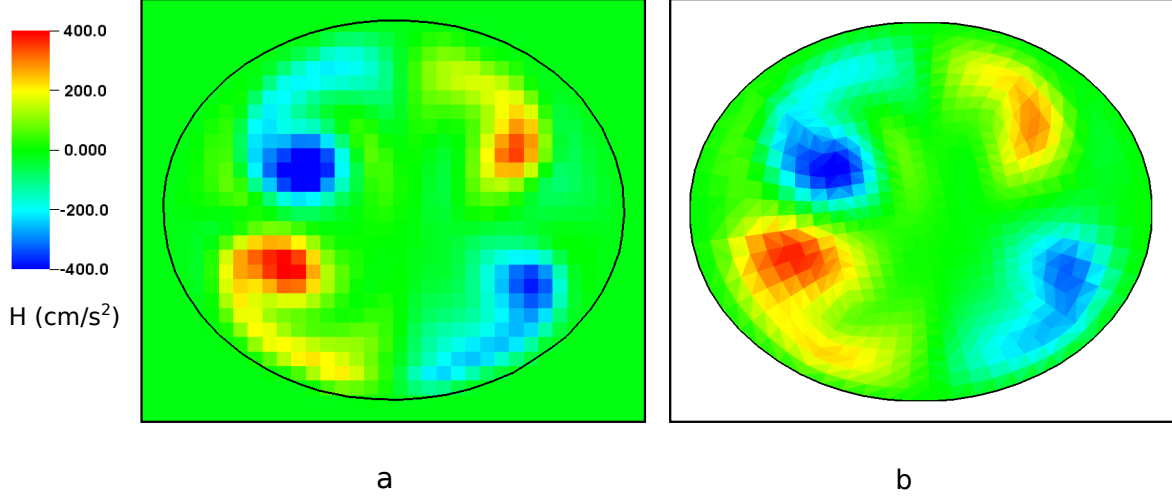


Figure 35: Local helicity density magnitudes in the IVC for (a) the present IIM and (b) the body-fitted OpenFOAM simulation.

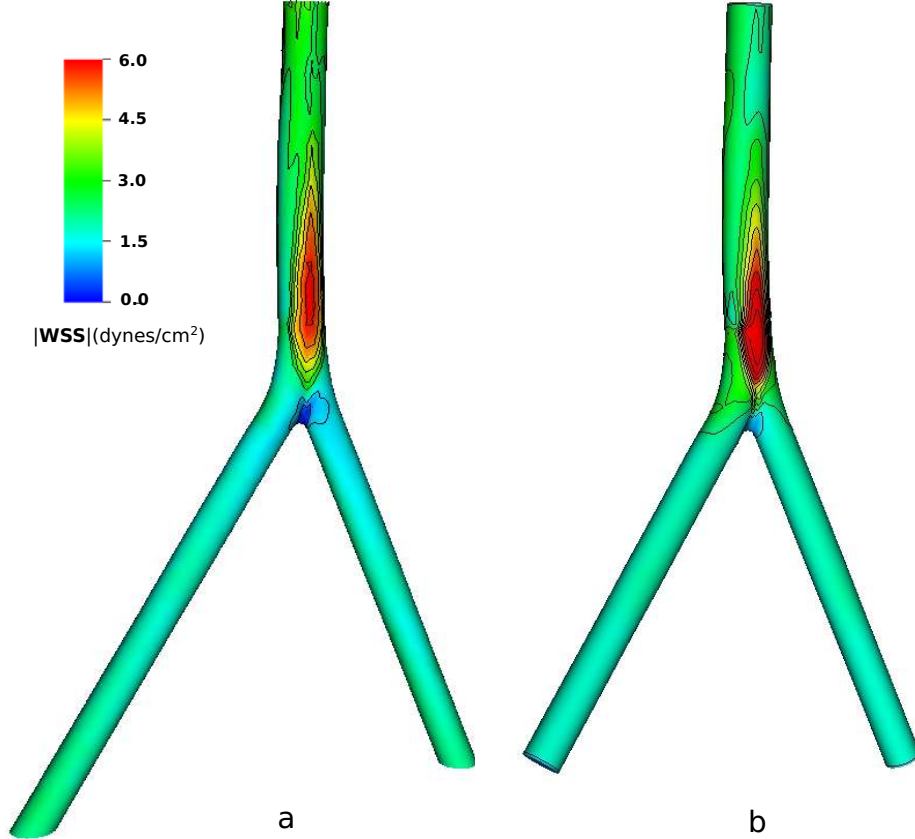


Figure 36: Contours of wall shear stress magnitude for flow in the IVC at the exercise flow rate for (a) the present IIM and (b) the OpenFOAM simulation.

agreement between the two solutions.

Finally, contours of wall shear stress magnitudes are plotted in Fig. 36 for both the present approach and the OpenFOAM simulation. The overall distribution appears to be similar particularly in the region prior

to the confluence of the veins and where the minimum wall shear stress occurs at the junction. However, slight differences are observed in the location where the maximum wall shear stress happens. This is likely due to minor differences in the mesh resolution between the two cases at the iliac vein confluence due to the requirement for point-matching in the block-structured body-fitted OpenFOAM mesh, which yields a slightly coarser resolution than the IIM mesh in this region. Future work should more rigorously compare the two simulation methods by performing a full mesh refinement study, which is beyond the scope of the present work.

5 Summary and conclusions

This work describes an immersed interface method that sharply resolves fluid dynamics problems in complex geometries using a general finite element mesh description of the interface. Like the IB method, the IIM does not require geometrically conforming spatial discretizations, which enables the use of fast Cartesian grid linear solvers. The primary contribution of the present work is to extend the IIM to treat general interface geometries that are described only in terms of C^0 -continuous surface representations. In particular, the present method does not require any analytic description of the interface geometry. Methods for accurately modeling flows involving C^0 -continuous immersed boundaries described by standard finite element meshes, without analytic information about the interfacial geometry, are needed to address applications involving geometries determined from experimental or clinical image data.

As in the original immersed interface methods, jumps in the pressure and velocity gradient are determined from interfacial forces imposed along the immersed boundary. In this work, a continuous representation of these jump conditions is determined on the C^0 representation of the interface by projecting them onto continuous Lagrangian basis functions supported along the interface. Although this requires the solution of a linear system of equations, this solve occurs only on the surface mesh and can be performed in an algorithmically optimal way using simple iterative methods. Further, an accurate velocity interpolation scheme based on a modified bilinear (or, in three spatial dimensions, trilinear) scheme is developed that accounts for the imposed discontinuity in the velocity gradient. Although the method uses only the lowest-order jump conditions, it is demonstrated to maintain global second-order accuracy in the Eulerian velocity and between first- and second-order global accuracy in the Eulerian pressure along with second-order accuracy in the pointwise deformation and velocity of the interface, and at least first-order accuracy in the pointwise fluid stress along the interface. Benchmark examples in two and three spatial dimensions demonstrate that very high fidelity flow simulations are possible using this scheme and, further, that it is necessary to account for all of these lowest-order jump conditions to maximize the accuracy of the scheme.

Another novel aspect of the present method is that it uses only the lowest-order jump conditions. It is common in immersed interface methods described in the literature to impose higher-order jumps conditions, including the jump in the first normal derivative of the pressure and the second normal derivative of the velocity. These higher-order corrections require the evaluation of terms such as the curvature and surface divergence of the force. Note that directly evaluating the curvature on the surface in a pointwise fashion requires C^2 continuity. C^0 -continuous surfaces do not provide enough regularity to obtain continuity of even the normal vector between elements. To address this difficulty, specialized treatments have been developed that yield accurate approximations of higher-order geometrical quantities, including the surface normal and curvature, on C^0 continuous surface representations. For triangulated surfaces, one approach is to adopt alternative basis functions, such as those used by subdivision surfaces [110], which provide enough regularity to yield accurate pointwise approximations to curvature and surface derivatives by directly differentiating the shape functions. Another approach is to use methods of discrete differential geometry using averaging Voronoi cells [111] or finite element stabilization techniques [112]. In the immersogeometric FSI methodology [113], higher order spline-based surface representation can be used, following the original work on isogeometric analysis [114]. Nonetheless, if global second-order accuracy in the velocity and first-order accuracy in the pressure and wall shear stress are sufficient, empirical tests reported in this paper demonstrate that only the lowest-order jump conditions are required. Further, these results demonstrate that these jump conditions may be determined and imposed using simple, low continuity-order interface representations. Work is underway to extend the present method to volumetric (codimension zero) problems of moving and flexible objects as well as thin flexible structures described using plate and shell models, and to apply this methodology to a

range of applications in engineering, applied science, and medicine.

Acknowledgement

We gratefully acknowledge research support through NIH Awards HL117063 and HL143336, NSF Awards DMS 1664645, CBET 175193, OAC 1450327, and OAC 1652541, and the U.S. FDA Center for Devices and Radiological Health (CDRH) Critical Path program. The research was supported in part by an appointment to the Research Participation Program at the U.S. FDA administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and FDA. A.P.S.B. also acknowledges support from San Diego State University. Computations were performed using facilities provided by University of North Carolina at Chapel Hill through the Research Computing division of UNC Information Technology Services and the high-performance computing clusters at the U.S. FDA. The findings and conclusions in this article have not been formally disseminated by the FDA and should not be construed to represent any agency determination or policy. The mention of commercial products, their sources, or their use in connection with material reported herein is not to be construed as either an actual or implied endorsement of such products by the Department of Health and Human Services. We thank Simone Rossi and Jae Ho Lee for their constructive comments to improve the manuscript. We also thank Kenneth Aycock for providing the patient-averaged IVC geometry.

References

- [1] C. S. Peskin, Flow patterns around heart valves: a numerical method, *J Comput Phys* 10 (2) (1972) 252–271.
- [2] C. S. Peskin, Numerical analysis of blood flow in the heart, *J Comput Phys* 25 (3) (1977) 220–252.
- [3] M.-C. Lai, Z. Li, A remark on jump conditions for the three-dimensional Navier–Stokes equations involving an immersed moving membrane, *Appl Math Lett* 14 (2) (2001) 149–154.
- [4] M.-C. Lai, C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J Comput Phys* 160 (2) (2000) 705–719.
- [5] B. E. Griffith, C. S. Peskin, On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems, *J Comput Phys* 208 (1) (2005) 75–105.
- [6] B. E. Griffith, R. D. Hornung, D. M. McQueen, C. S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, *J Comput Phys* 223 (1) (2007) 10–49.
- [7] B. E. Griffith, X. Luo, D. M. McQueen, C. S. Peskin, Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method, *Int J Appl Mech* 1 (01) (2009) 137–177.
- [8] B. E. Griffith, Immersed boundary model of aortic heart valve dynamics with physiological driving and loading conditions, *Int J Numer Method Biomed Eng* 28 (3) (2012) 317–345.
- [9] X. Y. Luo, B. E. Griffith, X. S. Ma, M. Yin, T. J. Wang, C. L. Liang, P. N. Watton, G. M. Bernacca, Effect of bending rigidity in a dynamic model of a polyurethane prosthetic mitral valve, *Biomech Model Mechanobiol* 11 (6) (2012) 815–827.
- [10] H. Gao, D. Carrick, C. Berry, B. E. Griffith, X. Y. Luo, Dynamic finite-strain modelling of the human left ventricle in health and disease using an immersed boundary-finite element method, *IMA J Appl Math* 79 (5) (2014) 978–1010.
- [11] V. Flamini, A. DeAnda, B. E. Griffith, Immersed boundary-finite element model of fluid-structure interaction in the aortic root, *Theor Comput Fluid Dynam* 30 (1) (2016) 139–164.
- [12] W. W. Chen, H. Gao, X. Y. Luo, N. A. Hill, Study of cardiovascular function using a coupled left ventricle and systemic circulation model, *J Biomech* 49 (12) (2016) 2445–2454.

- [13] H. Gao, L. Feng, N. Qi, C. Berry, B. E. Griffith, X. Y. Luo, A coupled mitral valve-left ventricle model with fluid-structure interaction, *Med Eng Phys* 47 (2017) 128–136.
- [14] A. Hasan, E. M. Kolahdouz, A. Enquobahrie, T. G. Caranasos, J. P. Vavalle, B. E. Griffith, Image-based immersed boundary model of the aortic root, *Med Eng Phys* 47 (2017) 72–84.
- [15] L. Y. Feng, N. Qi, H. Gao, W. Sun, M. Vazquez, B. E. Griffith, X. Y. Luo, On the chordae structure and dynamic behaviour of the mitral valve, *IMA J Appl MathEpub* ahead of print.
- [16] T. Skorczewski, B. E. Griffith, A. L. Fogelson, Multi-bond models for platelet adhesion and cohesion, in: S. D. Olson, A. T. Layton (Eds.), *Biological Fluid Dynamics: Modeling, Computation, and Applications*, Contemporary Mathematics, American Mathematical Society, Providence, RI, USA, 2014, pp. 149–172.
- [17] W. Kou, A. P. S. Bhalla, B. E. Griffith, J. E. Pandolfino, P. J. Kahrilas, N. A. Patankar, A fully resolved active musculo-mechanical model for esophageal transport, *J Comput Phys* 298 (2015) 446–465.
- [18] W. Kou, B. E. Griffith, J. E. Pandolfino, P. J. Kahrilas, N. A. Patankar, A continuum mechanics-based musculo-mechanical model for esophageal transport, *J Comput Phys* 348 (2017) 433–459.
- [19] W. Kou, J. E. Pandolfino, P. J. Kahrilas, N. A. Patankar, Studies of abnormalities of the lower esophageal sphincter during esophageal emptying based on a fully-coupled bolus-esophageal-gastric model, *Biomech Model Mechanobiol* 17 (4) (2018) 1069–1082.
- [20] L. Waldrop, L. Miller, Large-amplitude, short-wave peristalsis and its implications for transport, *Biomech Model Mechanobiol* 15 (3) (2016) 629–642.
- [21] N. A. Battista, A. N. Lane, J. Liu, L. A. Miller, Fluid dynamics of heart development: Effects of trabeculae and hematocrit, *Math Med Biol* 35 (4) (2018) 493–516, epub ahead of print.
- [22] S. K. Jones, R. Laurenza, T. L. Hedrick, B. E. Griffith, L. A. Miller, Lift vs. drag based mechanisms for vertical force production in the smallest flying insects, *J Theor Biol* 384 (2015) 105–120.
- [23] A. Santhanakrishnan, S. Jones, W. Dickson, M. Peek, V. Kasoju, M. Dickinson, L. Miller, Flow structure and force generation on flapping wings at low reynolds numbers relevant to the flight of tiny insects, *Fluids* 3 (3) (2018) 45.
- [24] S. Alben, L. A. Miller, J. Peng, Efficient kinematics for jet-propelled swimming, *J Fluid Mech* 733 (2013) 100–133.
- [25] H. Nguyen, L. Fauci, Hydrodynamics of diatom chains and semiflexible fibres, *J R Soc Interface* 11 (96) (2014) 20140314.
- [26] A. P. S. Bhalla, R. Bale, B. E. Griffith, N. A. Patankar, Fully resolved immersed electrohydrodynamics for particle motion, electrolocation, and self-propulsion, *J Comput Phys* 256 (2014) 88–108.
- [27] E. D. Tytell, C.-Y. Hsu, L. J. Fauci, The role of mechanical resonance in the neural control of swimming in fishes, *Zoology* 117 (1) (2014) 48–56.
- [28] R. Bale, A. P. S. Bhalla, I. D. Neveln, M. A. MacIver, N. A. Patankar, Convergent evolution of mechanically optimal locomotion in aquatic invertebrates and vertebrates, *PLOS Biol* 3 (4) (2015) e1002123.
- [29] A. P. Hoover, B. E. Griffith, L. A. Miller, Quantifying performance in the medusan mechanospace with an actively swimming three-dimensional jellyfish model, *J Fluid Mech* 813 (2017) 1112–1155.
- [30] N. Nangia, R. Bale, N. Chen, Y. Hanna, N. A. Patankar, Optimal specific wavelength for maximum thrust production in undulatory propulsion, *PLoS ONE* 12 (6) (2017) e0179727.
- [31] R. J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J Numer Anal* 31 (4) (1994) 1019–1044.

- [32] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, *J Comput Phys* 171 (2) (2001) 822–842.
- [33] L. Lee, R. J. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, *SIAM J Sci Comput* 25 (3) (2003) 832–856.
- [34] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J Comput Phys* 152 (2) (1999) 457–492.
- [35] R. P. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method, *J Comput Phys* 175 (1) (2002) 200–224.
- [36] M. Berger, R. Leveque, An adaptive cartesian mesh algorithm for the euler equations in arbitrary geometries, in: 9th Computational Fluid Dynamics Conference, 1989, p. 1930.
- [37] M. Berger, M. Aftosmis, Aspects (and aspect ratios) of cartesian mesh methods, in: Sixteenth International Conference on Numerical Methods in Fluid Dynamics, Springer, 1998, pp. 1–12.
- [38] D. K. Clarke, H. Hassan, M. Salas, Euler calculations for multielement airfoils using Cartesian grids, *AIAA journal* 24 (3) (1986) 353–358.
- [39] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J Comput Phys* 156 (2) (1999) 209–240.
- [40] M. F. Barad, P. Colella, S. G. Schladow, An adaptive cut-cell method for environmental fluid mechanics, *Int J Numer Methods Fluids* 60 (5) (2009) 473–514.
- [41] J. Mohd-Yusof, For simulations of flow in complex geometries, *Annual Research Briefs* 317.
- [42] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J Comput Phys* 161 (1) (2000) 35–60.
- [43] I. Borazjani, L. Ge, F. Sotiropoulos, Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies, *J Comput Phys* 227 (16) (2008) 7587–7620.
- [44] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies, *J Comput Phys* 207 (2) (2005) 457–492.
- [45] Y.-C. Chang, T. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, *J Comput Phys* 124 (2) (1996) 449–464.
- [46] R. Cortez, M. Minion, The blob projection method for immersed boundary problems, *J Comput Phys* 161 (2) (2000) 428–453.
- [47] S. Xu, Z. J. Wang, Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation, *SIAM J Sci Comput* 27 (6) (2006) 1948–1980.
- [48] R. J. LeVeque, Z. Li, Immersed interface methods for stokes flow with elastic boundaries or surface tension, *SIAM J Sci Comput* 18 (3) (1997) 709–735.
- [49] Z. Li, K. Ito, M.-C. Lai, An augmented approach for stokes equations with a discontinuous viscosity and singular forces, *Comput Fluids* 36 (3) (2007) 622–635.
- [50] D.-V. Le, B. C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *J Comput Phys* 220 (1) (2006) 109–138.
- [51] T. Y. Hou, Z. Li, S. Osher, H. Zhao, A hybrid method for moving interface problems with application to the hele–shaw flow, *J Comput Phys* 134 (2) (1997) 236–252.
- [52] J. A. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface methods, *J Comput Phys* 163 (2) (2000) 489–528.

- [53] J.-J. Xu, Z. Li, J. Lowengrub, H. Zhao, A level-set method for interfacial flows with surfactant, *J Comput Phys* 212 (2) (2006) 590–616.
- [54] S. Xu, Z. J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, *J Comput Phys* 216 (2) (2006) 454–493.
- [55] S. Xu, Z. J. Wang, A 3D immersed interface method for fluid–solid interaction, *Comput Methods Appl Mech Eng* 197 (25) (2008) 2068–2086.
- [56] Z. Li, K. Ito, *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*, SIAM, 2006.
- [57] T. Beale, A. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces, *Comm App Math Comp Sci* 1 (1) (2007) 91–119.
- [58] B. Lombard, J. Piriaux, Numerical treatment of two-dimensional interfaces for acoustic and elastic waves, *J Comput Phys* 195 (1) (2004) 90–116.
- [59] A. J. Bergou, S. Xu, Z. J. Wang, Passive wing pitch reversal in insect flight, *J Fluid Mech* 591 (2007) 321–337.
- [60] P. Jayathilake, Z. Tan, B. Khoo, N. Wijesundera, Deformation and osmotic swelling of an elastic membrane capsule in stokes flows by the immersed interface method, *Chem Eng Sci* 65 (3) (2010) 1237–1252.
- [61] A. El Yacoubi, S. Xu, Z. J. Wang, Computational study of the interaction of freely moving particles at intermediate Reynolds numbers, *J Fluid Mech* 705 (2012) 134–148.
- [62] E. M. Kolahdouz, D. Salac, A numerical model for the trans-membrane voltage of vesicles, *Appl Math Lett* 39 (2015) 7–12.
- [63] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J Comput Phys* 105 (2) (1993) 354–366.
- [64] B. Kallemov, A. Bhalla, B. Griffith, A. Donev, An immersed boundary method for rigid bodies, *Comm App Math Comp Sci* 11 (1) (2016) 79–141.
- [65] F. Balboa Usabiaga, B. Kallemov, B. Delmotte, A. Bhalla, B. Griffith, A. Donev, Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach, *Comm App Math Comp Sci* 11 (2) (2017) 217–296.
- [66] B. E. Griffith, On the volume conservation of the immersed boundary method, *Commun Comput Phys* 12 (2) (2012) 401–432.
- [67] B. E. Griffith, An accurate and efficient method for the incompressible Navier–Stokes equations using the projection method as a preconditioner, *J Comput Phys* 228 (20) (2009) 7565–7595.
- [68] W. J. Rider, J. A. Greenough, J. R. Kamm, Accurate monotonicity-and extrema-preserving methods through adaptive nonlinear hybridizations, *J Comput Phys* 225 (2) (2007) 1827–1848.
- [69] P. Colella, P. R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, *J Comput Phys* 54 (1) (1984) 174–201.
- [70] T. Möller, B. Trumbore, Fast, minimum storage ray/triangle intersection, in: *ACM SIGGRAPH 2005 Courses*, ACM, 2005, p. 7.
- [71] A. Lagae, P. Dutré, An efficient ray-quadrilateral intersection test, *Journal of Graphics Tools* 10 (4) (2005) 23–32.

- [72] Z. Tan, D.-V. Le, Z. Li, K. Lim, B. C. Khoo, An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane, *J Comput Phys* 227 (23) (2008) 9955–9983.
- [73] Z. Tan, D.-V. Le, K. Lim, B. Khoo, An immersed interface method for the incompressible Navier–Stokes equations with discontinuous viscosity across the interface, *SIAM J Sci Comput* 31 (3) (2009) 1798–1819.
- [74] B. E. Griffith, X. Y. Luo, Hybrid finite difference/finite element version of the immersed boundary method, *Int J Numer Meth Biomed Eng* 33 (11) (2017) e2888 (31 pages).
- [75] IBAMR: An adaptive and distributed-memory parallel implementation of the immersed boundary method, <https://github.com/IBAMR/IBAMR>.
- [76] SAMRAI: Structured adaptivemesh refinement application infrastructure, <http://www.llnl.gov/CASC/SAMRAI>.
- [77] R. D. Hornung, S. R. Kohn, Managing application complexity in the samrai object-oriented framework, *Concurr Comput* 14 (5) (2002) 347–368.
- [78] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc Web page, <http://www.mcs.anl.gov/petsc> (2018).
URL <http://www.mcs.anl.gov/petsc>
- [79] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.9, Argonne National Laboratory (2018).
URL <http://www.mcs.anl.gov/petsc>
- [80] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), *Modern Software Tools in Scientific Computing*, Birkhäuser Press, 1997, pp. 163–202.
- [81] hypre: High performance preconditioners, <http://www.llnl.gov/CASC/hypre>.
- [82] R. D. Falgout, U. M. Yang, hypre: A library of high performance preconditioners, in: *International Conference on Computational Science*, Springer, 2002, pp. 632–641.
- [83] libMesh: A C++ finite element library, <http://libmesh.github.io>.
- [84] B. S. Kirk, J. W. Peterson, R. H. Stogner, G. F. Carey, libmesh: A c++ library for parallel adaptive mesh refinement/coarsening simulations, *Eng Comput* 22 (3) (2006) 237–254.
- [85] T. G. Fai, C. H. Rycroft, Lubricated immersed boundary method in two dimensions, *Journal of Computational Physics* 356 (2018) 319–339.
- [86] H. Christensen, K. Tonder, The hydrodynamic lubrication of rough bearing surfaces of finite width, *Journal of Lubrication Technology* 93 (3) (1971) 324–329.
- [87] K. Taira, T. Colonius, The immersed boundary method: a projection approach, *Journal of Computational Physics* 225 (2) (2007) 2118–2137.
- [88] D. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, *J Fluid Mech* 6 (04) (1959) 547–567.
- [89] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, *J Comput Phys* 176 (2) (2002) 231–275.

- [90] M. Braza, P. Chassaing, H. H. Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, *J Fluid Mech* 165 (1986) 79–130.
- [91] C. Liu, X. Zheng, C. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J Comput Phys* 139 (1) (1998) 35–57.
- [92] A. Roshko, Experiments on the flow past a circular cylinder at very high Reynolds number, *J Fluid Mech* 10 (03) (1961) 345–356.
- [93] C. H. Williamson, Vortex dynamics in the cylinder wake, *Annu Rev Fluid Mech* 28 (1) (1996) 477–539.
- [94] D. Ingham, T. Tang, A numerical investigation into the steady flow past a rotating circular cylinder at low and intermediate Reynolds numbers, *J Comput Phys* 87 (1) (1990) 91–107.
- [95] K. I. Aycock, R. L. Campbell, K. B. Manning, B. A. Craven, A resolved two-way coupled CFD/6-DOF approach for predicting embolus transport and the embolus-trapping efficiency of ivc filters, *Biomech Model Mechanobiol* 16 (3) (2017) 851–869.
- [96] H. Badr, S. Dennis, P. Young, Steady and unsteady flow past a rotating circular cylinder at low Reynolds numbers, *Comput Fluids* 17 (4) (1989) 579–609.
- [97] S. D’alessio, S. Dennis, A vorticity model for viscous flow past a cylinder, *Comput Fluids* 23 (2) (1994) 279–293.
- [98] S. Mittal, B. Kumar, Flow past a rotating cylinder, *J Fluid Mech* 476 (2003) 303–334.
- [99] B. Fornberg, Steady viscous flow past a sphere at high Reynolds numbers., *J Fluid Mech* 190 (1) (1988) 471.
- [100] R. Turton, O. Levenspiel, A short note on the drag correlation for spheres, *Powder Technol* 47 (1) (1986) 83–86.
- [101] R. Campregher, J. Militzer, S. S. Mansur, A. d. Silveira Neto, Computations of the flow past a still sphere at moderate Reynolds numbers using an immersed boundary method, *J Braz Soc Mech Sci & Eng* 31 (4) (2009) 344–352.
- [102] J. C. Hunt, A. A. Wray, P. Moin, Eddies, streams, and convergence zones in turbulent flows.
- [103] H. Sakamoto, H. Haniu, A study on vortex shedding from spheres in a uniform flow, *J Fluids Eng* 112 (4) (1990) 386–392.
- [104] T. Johnson, V. Patel, Flow past a sphere up to a Reynolds number of 300, *J Fluid Mech* 378 (1999) 19–70.
- [105] E. Rahbar, D. Mori, J. E. Moore Jr, Three-dimensional analysis of flow disturbances caused by clots in inferior vena cava filters, *J Vasc Interv Radiol* 22 (6) (2011) 835–842.
- [106] B. A. Craven, K. I. Aycock, K. B. Manning, Steady flow in a patient-averaged inferior vena cava: Part i: computational fluid dynamics verification and validation, *Cardiovasc Eng Technol* (2018) 1–20.
- [107] M. B. Gallagher, K. I. Aycock, B. A. Craven, K. B. Manning, Steady flow in a patient-averaged inferior vena cava part i: particle image velocimetry measurements at rest and exercise conditions, *Cardiovascular engineering and technology* 9 (4) (2018) 641–653.
- [108] The CUBIT geometry and mesh generation toolkit (2007) sandia national laboratories, <http://cubit.sandia.gov>.
- [109] D. Mukherjee, J. Padilla, S. C. Shadden, Numerical investigation of fluid–particle interactions for embolic stroke, *Theoretical and Computational Fluid Dynamics* 30 (1-2) (2016) 23–39.
- [110] F. Cirak, M. Ortiz, P. Schröder, Subdivision surfaces: a new paradigm for thin-shell finite-element analysis, *Int J Numer Methods Eng* 47 (12) (2000) 2039–2072.

- [111] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, in: Visualization and mathematics III, Springer, 2003, pp. 35–57.
- [112] M. Cenanovic, P. Hansbo, M. G. Larson, Finite element procedures for computing normals and mean curvature on triangulated surfaces and their use for mesh refinement, arXiv preprint arXiv:1703.05745.
- [113] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, T. J. Hughes, An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves, Computer methods in applied mechanics and engineering 284 (2015) 1005–1053.
- [114] T. J. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement, Computer methods in applied mechanics and engineering 194 (39-41) (2005) 4135–4195.

A Convergence data

Table 4: Convergence results of the Eulerian velocity and pressure for two-dimensional horizontal plane-Poiseuille considered in Sec. 4.1.1.

	h	velocity		pressure	
		Error	Order	Error	Order
L^2	1.56×10^{-1}	4.4886×10^{-2}	-	2.9764×10^{-2}	-
	7.80×10^{-2}	1.0044×10^{-2}	2.16	1.2007×10^{-2}	1.31
	3.90×10^{-2}	1.3822×10^{-3}	2.86	2.5519×10^{-3}	2.23
	1.95×10^{-3}	3.6201×10^{-4}	1.93	2.3269×10^{-4}	3.46
L^∞	1.56×10^{-1}	4.6171×10^{-2}	-	3.4157×10^{-2}	-
	7.80×10^{-2}	1.0396×10^{-2}	2.15	1.2673×10^{-2}	1.43
	3.90×10^{-2}	2.5911×10^{-3}	2.00	2.7193×10^{-3}	2.22
	1.95×10^{-3}	8.4275×10^{-4}	1.62	2.6682×10^{-4}	3.35

Table 5: Convergence results of the displacement, velocity, pressure and WSS on the surface mesh of the two-dimensional horizontal plane-Poiseuille flow considered in Sec. 4.1.1.

	h	displacement		velocity		pressure		WSS	
		Error	Order	Error	Order	Error	Order	Error	Order
L^2	1.56×10^{-1}	4.2943×10^{-3}	-	8.6485×10^{-3}	-	4.2904×10^{-2}	-	1.9004×10^{-2}	-
	7.80×10^{-2}	1.0702×10^{-3}	2.00	2.1488×10^{-3}	2.01	1.5327×10^{-2}	1.49	8.9271×10^{-3}	1.09
	3.90×10^{-2}	2.8847×10^{-4}	1.89	8.6136×10^{-4}	1.32	1.8084×10^{-3}	2.55	4.2621×10^{-3}	1.07
	1.95×10^{-3}	7.1420×10^{-5}	2.01	2.1852×10^{-4}	1.98	2.0321×10^{-4}	3.15	2.1631×10^{-3}	0.98
L^∞	1.56×10^{-1}	5.0088×10^{-3}	-	1.0645×10^{-2}	-	1.0489×10^{-2}	-	1.1273×10^{-2}	-
	7.80×10^{-2}	1.0452×10^{-3}	2.26	2.9337×10^{-3}	1.86	4.3378×10^{-3}	1.27	5.3873×10^{-3}	1.07
	3.90×10^{-2}	2.4787×10^{-4}	2.08	1.5121×10^{-3}	0.96	6.8581×10^{-4}	2.66	2.5205×10^{-3}	1.09
	1.95×10^{-3}	6.1995×10^{-5}	2.00	4.0972×10^{-4}	1.88	1.0069×10^{-4}	2.78	1.2724×10^{-3}	0.99

Table 6: Convergence results of the Eulerian velocity and pressure for two-dimensional inclined plane-Poiseuille flow considered in Sec. 4.1.1.

	h	velocity		pressure		pressure on Ω^*	
		Error	Order	Error	Order	Error	Order
L^2	7.80×10^{-2}	1.3072×10^{-3}	-	7.4462×10^{-4}	-	-	-
	3.90×10^{-2}	3.1417×10^{-4}	2.06	2.2671×10^{-4}	1.72	-	-
	1.95×10^{-3}	7.6507×10^{-5}	2.04	9.4735×10^{-5}	1.26	-	-
	9.75×10^{-4}	1.9933×10^{-5}	1.94	4.0041×10^{-5}	1.24	-	-
L^∞	7.80×10^{-2}	5.8251×10^{-2}	-	4.5021×10^{-2}	-	5.2985×10^{-2}	-
	3.90×10^{-2}	3.0388×10^{-2}	0.94	2.1777×10^{-2}	1.48	1.9759×10^{-2}	1.42
	1.95×10^{-3}	1.5299×10^{-2}	0.99	1.5221×10^{-2}	0.52	1.0058×10^{-2}	0.97
	9.75×10^{-4}	7.6174×10^{-3}	1.00	1.4029×10^{-2}	0.12	5.0311×10^{-3}	1.00

Table 7: Convergence results of the displacement, velocity, pressure and WSS on the surface mesh of the two-dimensional inclined plane-Poiseuille flow considered in Sec. 4.1.1.

	h	displacement		velocity		Pressure		WSS	
		Error	Order	Error	Order	Error	Order	Error	Order
L^2	7.80×10^{-2}	1.0256×10^{-3}	-	5.4813×10^{-3}	-	3.4355×10^{-2}	-	1.0341×10^{-2}	-
	3.90×10^{-2}	2.5688×10^{-4}	2.00	8.1599×10^{-4}	2.75	1.4422×10^{-2}	1.25	4.9935×10^{-2}	1.05
	1.95×10^{-3}	6.5622×10^{-5}	1.97	2.3934×10^{-4}	1.77	4.0782×10^{-3}	1.82	2.3804×10^{-2}	1.07
	9.75×10^{-4}	1.6489×10^{-5}	1.99	6.2818×10^{-5}	1.93	1.1990×10^{-3}	1.77	1.1239×10^{-3}	1.08
L^∞	7.80×10^{-2}	1.3363×10^{-3}	-	7.0881×10^{-3}	-	2.9897×10^{-2}	-	6.9130×10^{-3}	-
	3.90×10^{-2}	2.9123×10^{-4}	2.20	1.1809×10^{-3}	2.59	1.2979×10^{-2}	1.20	3.1308×10^{-3}	1.14
	1.95×10^{-3}	7.0620×10^{-5}	2.04	4.1602×10^{-4}	1.51	4.5329×10^{-3}	1.52	1.7871×10^{-3}	0.81
	9.75×10^{-4}	1.8191×10^{-5}	1.96	1.4445×10^{-4}	1.53	1.5480×10^{-3}	1.55	8.7958×10^{-4}	1.02

Table 8: Convergence results of the Eulerian velocity and pressure for three-dimensional horizontal Hagen-Poiseuille flow considered in Sec. 4.1.2.

	h	velocity		pressure		pressure on Ω^*	
		Error	Order	Error	Order	Error	Order
L^2	1.56×10^{-1}	1.1629×10^{-1}	-	6.0942×10^{-2}	-	-	-
	7.80×10^{-2}	2.9558×10^{-2}	1.98	1.6789×10^{-2}	1.86	-	-
	3.90×10^{-2}	7.6507×10^{-3}	2.04	2.4049×10^{-3}	2.80	-	-
	1.95×10^{-3}	1.7100×10^{-3}	2.07	6.4996×10^{-4}	1.89	-	-
L^∞	1.56×10^{-1}	4.8008×10^{-2}	-	1.1099×10^{-1}	-	1.1099×10^{-1}	-
	7.80×10^{-2}	2.3307×10^{-2}	1.04	6.9275×10^{-2}	0.68	2.4625×10^{-2}	2.17
	3.90×10^{-2}	4.7934×10^{-3}	2.28	4.4148×10^{-2}	0.65	3.0722×10^{-3}	3.00
	1.95×10^{-3}	1.2291×10^{-3}	1.96	2.5007×10^{-2}	0.82	7.5856×10^{-4}	2.02

Table 9: Convergence results of the displacement, velocity, pressure and WSS on the surface mesh of the three-dimensional horizontal Hagen-Poiseuille flow considered in Sec. 4.1.2.

	h	displacement		velocity		pressure		WSS	
		Error	Order	Error	Order	Error	Order	Error	Order
L^2	1.56×10^{-1}	5.8931×10^{-3}	-	4.4541×10^{-3}	-	1.7140×10^{-1}	-	2.3026×10^{-2}	-
	7.80×10^{-2}	8.5408×10^{-4}	2.79	4.8937×10^{-4}	3.19	4.1685×10^{-2}	2.04	1.2835×10^{-2}	1.09
	3.90×10^{-2}	2.1148×10^{-4}	2.01	4.0652×10^{-4}	0.26	5.2332×10^{-3}	2.99	5.8642×10^{-3}	1.07
	1.95×10^{-3}	5.2629×10^{-5}	2.00	9.9410×10^{-5}	2.03	1.2725×10^{-3}	2.04	2.7800×10^{-3}	0.98
L^∞	1.56×10^{-1}	4.6561×10^{-3}	-	8.5493×10^{-3}	-	1.1037×10^{-1}	-	1.4015×10^{-2}	-
	7.80×10^{-2}	7.4997×10^{-4}	2.63	1.6748×10^{-3}	2.35	2.4535×10^{-2}	1.27	7.5266×10^{-3}	0.90
	3.90×10^{-2}	1.5650×10^{-4}	2.26	1.0312×10^{-3}	0.70	3.0398×10^{-3}	2.66	3.5189×10^{-3}	1.10
	1.95×10^{-3}	3.9278×10^{-5}	1.99	3.0996×10^{-4}	1.73	7.2375×10^{-4}	2.78	1.7256×10^{-3}	1.03

Table 10: Convergence results of the Eulerian velocity and pressure for three-dimensional inclined Hagen-Poiseuille flow considered in Sec. 4.1.2.

	h	velocity		pressure		pressure on Ω^*	
		Error	Order	Error	Order	Error	Order
L^2	7.80×10^{-2}	2.7733×10^{-3}	-	6.5465×10^{-2}	-	-	-
	3.90×10^{-2}	4.6659×10^{-4}	2.57	1.8020×10^{-3}	1.86	-	-
	1.95×10^{-3}	1.8128×10^{-4}	1.36	1.0238×10^{-3}	0.82	-	-
	9.75×10^{-4}	4.5095×10^{-5}	2.01	3.6563×10^{-4}	1.49	-	-
L^∞	7.80×10^{-2}	8.4694×10^{-3}	-	4.5885×10^{-2}	-	3.9710×10^{-2}	-
	3.90×10^{-2}	2.2634×10^{-3}	1.90	4.0355×10^{-2}	0.19	5.7205×10^{-3}	2.80
	1.95×10^{-3}	7.9415×10^{-4}	1.51	2.1088×10^{-3}	0.94	2.1088×10^{-3}	1.44
	9.75×10^{-4}	2.2101×10^{-4}	1.85	1.4058×10^{-3}	0.59	1.0088×10^{-3}	1.06

Table 11: Convergence results of the displacement, velocity, pressure and WSS on the surface mesh of the three-dimensional inclined Hagen-Poiseuille flow considered in Sec. 4.1.2.

	h	displacement		velocity		pressure		WSS	
		Error	Order	Error	Order	Error	Order	Error	Order
L^2	7.80×10^{-2}	2.8315×10^{-3}	-	1.8397×10^{-5}	-	3.1258×10^{-3}	-	8.3374×10^{-3}	-
	3.90×10^{-2}	4.1330×10^{-4}	2.78	5.0375×10^{-6}	1.87	3.0608×10^{-4}	3.35	4.7784×10^{-3}	0.80
	1.95×10^{-3}	1.0007×10^{-4}	2.05	3.0499×10^{-6}	0.72	1.1570×10^{-4}	1.40	2.4453×10^{-3}	0.97
	9.75×10^{-4}	2.4407×10^{-5}	2.04	7.8403×10^{-7}	1.96	3.6156×10^{-5}	1.68	1.2870×10^{-3}	0.93
L^∞	7.80×10^{-2}	2.1752×10^{-3}	-	1.1652×10^{-5}	-	3.1373×10^{-3}	-	5.7894×10^{-2}	-
	3.90×10^{-2}	3.3874×10^{-4}	2.68	5.4306×10^{-6}	1.10	3.3588×10^{-4}	3.22	2.9846×10^{-3}	0.96
	1.95×10^{-3}	7.9366×10^{-5}	2.09	3.0997×10^{-6}	0.81	1.8660×10^{-4}	0.85	1.5948×10^{-3}	0.90
	9.75×10^{-4}	1.8852×10^{-5}	2.07	1.4285×10^{-6}	1.12	9.8211×10^{-5}	0.93	8.0262×10^{-3}	0.99

Table 12: Convergence results of the Eulerian velocity and pressure for two-dimensional circular Couette flow considered in Sec. 4.2.

	h	velocity		pressure		pressure on Ω^*	
		Error	Order	Error	Order	Error	Order
L^2	1.2500×10^{-1}	3.9912×10^{-2}	-	5.7864×10^{-2}	-	-	-
	6.2500×10^{-2}	1.0652×10^{-2}	1.91	1.6002×10^{-2}	1.85	-	-
	3.1250×10^{-2}	3.6850×10^{-3}	1.53	4.4463×10^{-3}	1.85	-	-
	1.5625×10^{-2}	9.5675×10^{-4}	1.95	1.6014×10^{-3}	1.74	-	-
L^∞	1.2500×10^{-1}	1.2378×10^{-1}	-	4.5941×10^{-2}	-	2.7895×10^{-2}	-
	6.2500×10^{-2}	4.0347×10^{-2}	1.62	4.0176×10^{-2}	0.19	1.3461×10^{-2}	1.05
	3.1250×10^{-2}	2.2022×10^{-2}	0.87	2.8997×10^{-2}	0.47	6.8975×10^{-3}	0.96
	1.5625×10^{-2}	5.7101×10^{-3}	1.95	1.8695×10^{-2}	0.63	3.4660×10^{-3}	0.99

Table 13: Convergence results of the displacement, velocity, pressure and WSS on the surface mesh of the two-dimensional circular Couette flow considered in Sec. 4.2.

	h	displacement		velocity		pressure		WSS	
		Error	Order	Error	Order	Error	Order	Error	Order
L^2	1.2500×10^{-1}	3.4268×10^{-3}	-	4.2827×10^{-2}	-	2.7572×10^{-1}	-	2.5266×10^{-2}	-
	6.2500×10^{-2}	8.5475×10^{-4}	2.00	1.2115×10^{-2}	1.82	6.7620×10^{-1}	2.03	1.4307×10^{-2}	0.82
	3.1250×10^{-2}	2.0304×10^{-4}	2.07	3.6922×10^{-3}	1.71	1.7609×10^{-2}	1.94	8.2638×10^{-3}	0.79
	1.5625×10^{-2}	5.0057×10^{-5}	2.02	9.3743×10^{-4}	1.98	6.4982×10^{-3}	1.44	4.3695×10^{-3}	0.92
L^∞	1.2500×10^{-1}	3.2031×10^{-3}	-	3.6900×10^{-2}	-	1.6974×10^{-1}	-	1.5876×10^{-2}	-
	6.2500×10^{-2}	6.7864×10^{-4}	2.24	1.1092×10^{-2}	1.73	4.7125×10^{-2}	1.85	8.3447×10^{-3}	0.93
	3.1250×10^{-2}	1.5153×10^{-4}	2.16	3.8568×10^{-3}	1.52	1.7521×10^{-2}	1.43	4.0669×10^{-3}	1.04
	1.5625×10^{-2}	3.4132×10^{-5}	2.15	9.3019×10^{-4}	2.05	8.4316×10^{-3}	1.06	2.2926×10^{-3}	0.83

Table 14: Convergence results of the Eulerian velocity and pressure for three-dimensional circular Couette flow considered in Sec. 4.2.

	h	velocity		pressure		pressure on Ω^*	
		Error	Order	Error	Order	Error	Order
L^2	1.2500×10^{-1}	7.9945×10^{-2}	-	1.6719×10^{-1}	-	-	-
	6.2500×10^{-2}	2.1306×10^{-2}	1.91	5.0511×10^{-2}	1.73	-	-
	3.1250×10^{-2}	7.3705×10^{-3}	1.53	1.1295×10^{-2}	2.16	-	-
	1.5625×10^{-2}	1.9527×10^{-3}	1.92	3.8657×10^{-3}	1.55	-	-
L^∞	1.2500×10^{-1}	1.0378×10^{-1}	-	4.6053×10^{-2}	-	3.6172×10^{-2}	-
	6.2500×10^{-2}	4.3354×10^{-2}	1.26	4.3662×10^{-2}	0.077	1.8408×10^{-2}	0.97
	3.1250×10^{-2}	2.0026×10^{-2}	1.11	3.1003×10^{-2}	0.49	9.4395×10^{-3}	0.96
	1.5625×10^{-2}	6.7114×10^{-3}	1.58	2.0668×10^{-2}	0.58	4.7551×10^{-3}	0.99

Table 15: Convergence results of the displacement, velocity, pressure and WSS on the surface mesh of the three-dimensional circular Couette flow considered in Sec. 4.2.

	h	displacement		velocity		pressure		WSS	
		Error	Order	Error	Order	Error	Order	Error	Order
L^2	1.2500×10^{-1}	6.8526×10^{-3}	-	8.5642×10^{-2}	-	1.16276×10^0	-	5.8980×10^{-2}	-
	6.2500×10^{-2}	1.7098×10^{-3}	2.00	2.6461×10^{-2}	1.69	2.8960×10^{-1}	2.01	3.1341×10^{-2}	0.91
	3.1250×10^{-2}	4.0606×10^{-4}	2.07	7.3760×10^{-3}	1.84	7.2821×10^{-2}	1.99	1.7820×10^{-2}	0.81
	1.5625×10^{-2}	1.0023×10^{-4}	2.02	1.8712×10^{-3}	1.98	2.5551×10^{-2}	1.51	9.4697×10^{-3}	0.91
L^∞	1.2500×10^{-1}	3.2012×10^{-3}	-	3.7025×10^{-2}	-	3.3877×10^{-1}	-	1.6843×10^{-2}	-
	6.2500×10^{-2}	6.7975×10^{-4}	2.24	1.1325×10^{-2}	1.71	8.9525×10^{-2}	1.92	9.7928×10^{-3}	0.78
	3.1250×10^{-2}	1.5146×10^{-4}	2.17	3.8613×10^{-3}	1.55	3.1973×10^{-2}	1.49	5.3490×10^{-3}	0.87
	1.5625×10^{-2}	3.4150×10^{-5}	2.15	9.3525×10^{-4}	2.04	1.5225×10^{-2}	1.07	2.8636×10^{-3}	0.90