Journal of Computational Physics 464 (2022) 111339

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An immersed interface method for the 2D vorticity-velocity Navier-Stokes equations with multiple bodies

James Gabbard [a], Thomas Gillis [a], Philippe Chatelain [b], Wim M. van Rees [a,*]

[a] Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[b] Institute of Mechanics, Materials and Civil Engineering, Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Article history:
Received 22 April 2021
Received in revised form 2 May 2022
Accepted 4 May 2022
Available online 1 June 2022

Keywords:
Immersed interface method
Vorticity formulation
Conservative finite difference
Discrete Kelvin's theorem

ABSTRACT

We present an immersed interface method for the vorticity-velocity form of the 2D Navier Stokes equations that directly addresses challenges posed by nonconvex immersed bodies, multiply connected domains, and the calculation of force distributions on immersed surfaces. The immersed interface method is re-interpreted as a polynomial extrapolation of flow quantities and boundary conditions into the immersed solid bodies, reducing computational cost and enabling simulations with nonconvex bodies that could not be discretized with previous immersed interface methods. In the flow, the vorticity transport equation is discretized using a conservative finite difference scheme and explicit Runge-Kutta time integration. The velocity reconstruction problem is transformed to a scalar Poisson equation that is discretized with conservative finite differences, and solved using an FFT-accelerated iterative algorithm. The use of conservative differencing throughout leads to exact enforcement of a discrete Kelvin's theorem, allowing for simulations with multiply connected domains and outflow boundaries that have challenged other immersed interface vortex methods. We also explore novel methods for recovering time-dependent pressure distributions on immersed bodies within a vorticity-based method and present a novel control volume formulation for recovering aerodynamic moments from only the vorticity and velocity fields. The method achieves second order spatial accuracy and third order temporal accuracy, and is validated on a variety of 2D flows in internal and free-space domains.

1. Introduction

Immersed methods solve partial differential equations inside or outside of irregular domains, while using a regular structured grid (typically Cartesian). The benefit of not having to adapt the underlying mesh to the domain boundaries provides simplicity and computational efficiency in handling complex domain geometries, arbitrary topologies (e.g. multiple immersed bodies), and dynamically moving domain boundaries. These characteristics are especially of interest when combined with the Navier-Stokes equations to solve flow problems such as biologically-inspired locomotion. Broadly, there are two classes of immersed methods for incompressible Navier-Stokes simulations [1]. Continuous forcing methods include traditional immersed boundary methods [2,1,3], and Brinkmann penalization [4–6]. These methods add a singular forcing term to the continuous Navier Stokes equations within solid regions, which approximately enforces a no-slip condition on

* Corresponding author.
E-mail address: wvanrees@mit.edu (W.M. van Rees).

https://doi.org/10.1016/j.jcp.2022.111339
0021-9991/

solid boundaries. To maintain regularity after discretization, the forcing term is either smoothed on the object boundary and its value is resolved dynamically [6], or an iterative process is used to enforce the boundary condition [7]. This limits many such methods to first-order spatial and temporal accuracy. Discrete forcing methods, on the other hand, include sharp immersed boundary methods [8,9], immersed interface methods [10,11], and other relatives such as Ghost Fluid [12], Ghost Cell [13], and cut cell finite volume methods [14]. These approaches use a modified discretization near solid objects that sharply resolves the location of immersed boundaries and enforces corresponding boundary conditions. Although these modifications are more challenging to derive and implement, they allow for increased spatial and temporal accuracy, as well as accurate resolution of local flow quantities such as traction forces on immersed solid boundaries.

Here we focus on the immersed interface method (IIM), a term that in itself covers a broad collection of discrete forcing methods. Early Navier-Stokes simulations used the IIM to discretize singular sources such as forcing terms representing interfaces with surface tension, or elastic membranes [15–17]. With the introduction of the explicit jump immersed interface method (EJIIM) [18], the IIM was extended from discretizing singular source terms to handle directly imposed boundary conditions such as Dirichlet or Neumann conditions. The EJIIM relies on the use of jump-corrected Taylor series within standard finite difference schemes, keeping the solution a linear combination of grid values while incorporating boundary conditions. Further, the method uses a dimensionally-split approach, simplifying its extension to 2D and 3D. Combined, the EJIIM and its newest iterations (including our work) overlap significantly with sharp immersed boundary methods, and have much in common with other methods (such as the Ghost Cell, Cut Cell, and Ghost Fluid Methods). Since the development of the EJIIM, immersed interface methods have expanded to allow more complex PDEs, higher-order discretizations, and provably stable boundary treatments. In particular, we highlight recent developments in GKS-stable high-order boundary treatments for hyperbolic PDEs [19,20], parabolic PDEs [21,22], and advection-diffusion equations [23] with irregular domain boundaries; local stability optimizations for advection-dominant PDEs with immersed boundaries [24–26]; and several treatments of moving immersed boundaries that allow for at least second order accuracy [17,27,28].

Around the early 2000s, the IIM was combined for the first time with vorticity-based formulations of the 2D Navier-Stokes equations [29,30], which used similar jump-corrected finite difference schemes as the EJIIM. These works are characterized by a temporal splitting approach to solve the Stokes problem with appropriate global vorticity boundary conditions, providing consistent second-order spatial accuracy and first-order temporal accuracy. In Linnick and Fasel [31], the authors employ the EJIIM to provide a 2D vorticity-velocity Navier-Stokes solver with compact difference schemes and a Thom-like, local vorticity boundary condition that enabled fourth-order spatial and temporal accuracy. This approach was recently extended using a more efficient multigrid solver in [32], and implemented in 3D in [33]. Recently, the IIM has also been integrated within vortex particle-mesh methods, which rely on a combined Lagrangian-Eulerian approach to integrate the incompressible Navier-Stokes equations. In [34] a traditional Lighthill splitting approach was introduced to handle the vorticity boundary condition, leading to first-order temporal and second-order spatial accuracy. Subsequently, the same group employed a Thom-like finite difference boundary condition to achieve high-order accuracy in time [35] and an extension to 3D [36]. These latter results used a Lattice Green's Functions FFT-accelerated Poisson solver, together with a Schur-complement boundary approach, solved using recycling GMRes [37,38].

The majority of the approaches above handle the external flow around a single, stationary, and typically convex object. A major challenge in extending towards multiple bodies is the need to enforce circulation conservation on each body independently. Here we build off the approaches in [34,35] to develop a vorticity-based 2D finite-difference IIM that addresses this issue, extending these methods to simulations with multiple immersed bodies, nonconvex immersed bodies, and outflow boundary conditions. To do so we use a conservative finite-difference discretization of the Navier-Stokes equations that allows us to explicitly enforce circulation conservation in the presence of multiple bodies and outflows. This method is shown to be second-order accurate in space and third-order accurate in time, and it achieves excellent computational efficiency through the use of a fast IIM elliptic solver developed in [38]. Further, we consider novel methods for the calculation of time-dependent pressure and shear distributions on immersed surfaces within a vorticity-velocity based IIM. More broadly, we provide a novel interpretation of the EJIIM through the lens of ghost points reconstructed with a polynomial extrapolation, which greatly simplifies implementation details and exposes how the EJIIM is related to sharp-interface immersed-boundary and finite-volume methods.

The rest of this work is structured as follows. Section 2 introduces the EJIIM and its relation to ghost point reconstruction, as well as its application to nonconvex bodies in 2D. Sections 3 and 4 discuss IIM discretizations of the vorticity transport equation and the elliptic velocity reconstruction problem, respectively. In section 5 these two discretizations are combined into a full Navier-Stokes discretization which enforces a discrete form of Kelvin's theorem. Section 5.4 introduces techniques for calculating forces and surface tractions acting on immersed bodies, which are applied to a variety of flows in section 6 to illustrate the accuracy and effectiveness of the methods presented here. We conclude in section 7 with a summary of our contributions and a discussion of future directions for this work.

2. The immersed interface method

In this section we briefly review the explicit jump immersed interface method (EJIIM) in a 1D setting and discuss a specialization of the method which reduces complexity and enhances numerical stability. This specialization is then extended to 2D problems.

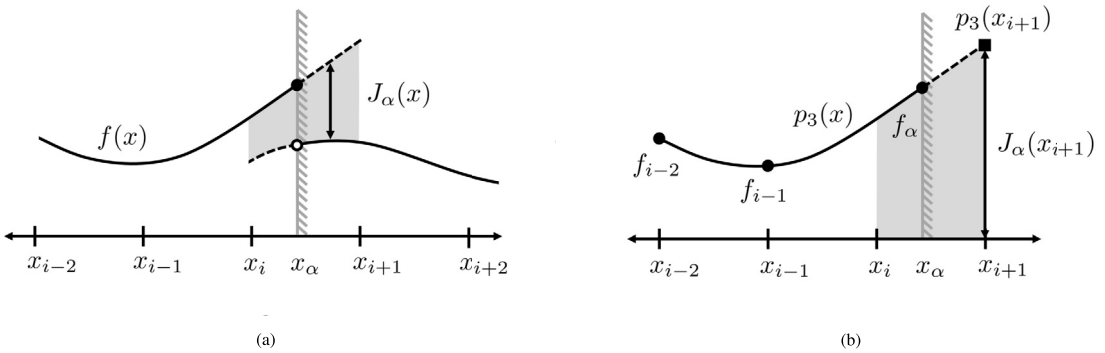


Fig. 1. Two use cases for the Explicit Jump IIM. (a) For a physical interface, the solution $f(x)$ has nontrivial dynamics on both sides of a discontinuity. The jump in $f(x)$ and its derivatives are known a priori from physical principles. (b) For a domain boundary, the solution is trivial on one side of the discontinuity, and a Dirichlet boundary condition is prescribed. However, there is no information on the jump in the derivatives.

2.1. Specializing the explicit jump IIM

The explicit jump immersed interface method (EJIIM), introduced by Wiegmann and Bube [18], is a method of adapting regular finite difference schemes for equations with discontinuous solutions. At its center is the construction of modified Taylor series expansions which correctly approximate functions with jump discontinuities. To illustrate, consider a function $f(x)$ that is smooth except at a point x_α, where there is a jump singularity in $f(x)$ and its derivatives $f^{(k)}(x)$. Let $(f^{(k)})^-$ and $(f^{(k)})^+$ denote the value of $f^{(k)}$ on the left and right sides of the discontinuity, respectively, and let $[f^{(k)}]_\alpha = (f^{(k)})^+ - (f^{(k)})^-$ denote the magnitude of the jump in $f^{(k)}$ at x_α. Finally, consider a regular grid of points $x_i = ih$, with the point x_α contained in the interval $[x_i, x_{i+1}]$ (Fig. 1a). Given the values of $f^{(k)}(x_i)$ and $[f^{(k)}]_\alpha$, the function f can be extrapolated from x_i to x_{i+1} using the modified Taylor series

$$f(x_{i+1}) = \sum_{k=0}^{n} \frac{h^k}{k!} f^{(k)}(x_i) + J_\alpha(x_{i+1}) + O(h^{n+1}), \quad \text{with} \quad J_\alpha(x_{i+1}) = \sum_{k=0}^{n} \frac{(x_{i+1} - x_\alpha)^k}{k!} [f^{(k)}]_\alpha. \tag{1}$$

The first half of (1) is a standard Taylor expansion of f about x_i; the second is a jump correction that must be added to any expansion that crosses the discontinuity. In the EJIIM, these generalized Taylor series are used to construct jump-corrected finite difference stencils which retain their high-order accuracy across the jump discontinuity at x_α. This method is well suited to physical interfaces where the jumps $[f^{(k)}]_\alpha$ are determined by the geometry of the interface or a known discontinuity in a prescribed source field.

Since its original publication, the EJIIM has been repeatedly re-purposed to discretize problems with smooth solutions that are posed on irregular domains. To tackle such problems, a common approach is to prescribe the solution outside of the problem domain (typically to zero value), and then treat the irregular domain boundary as a jump discontinuity. In this case the jump discontinuity is no longer physically constrained, and the value of the jump in each derivative $f^{(k)}(x_\alpha)$ must be calculated directly from the function $f(x)$ by evaluating a one-sided finite difference stencil [31,35]. To illustrate this procedure, consider the same function f discussed above, now with $f(x) = 0$ for $x > x_\alpha$ to model a domain boundary (Fig. 1b). Given the value of $f(x)$ on the regular grid and a Dirichlet boundary condition $f(x_\alpha)$, the interface derivatives $(f^{(k)})^-$ are approximated by the $n+1$ point one-sided finite differences

$$(f^{(k)})^-_{FD} = S^k_\alpha f(x_\alpha) + \sum_{j=1}^{n} S^k_j f(x_{i-j}) = (f^{(k)})^- + O\Big(h^{n+1-k}\Big). \tag{2}$$

Here the point x_i has been excluded from the stencil to avoid ill conditioning when $|x_\alpha - x_i|$ is small. The optimal order of accuracy in (2) is achieved by taking $S^{(k)}_j = \ell^{(k)}_j(x_\alpha)$, where the $\ell_j(x)$ are the degree n Lagrange polynomials satisfying $\ell_i(x_j) = \delta_{ij}$ for $x_j \in \{x_\alpha,\ x_{i-1},\ x_{i-2},\ \ldots,\ x_{i-n}\}$. Using the approximation of $(f^{(k)})^-$ given in (2) and taking $(f^{(k)})^+ = 0$ leads to the approximate jump correction

$$J_\alpha(x_{i+1}) = -\sum_{k=0}^{n} \frac{(x_{i+1} - x_\alpha)^k}{k!} (f^{(k)})^-_{FD} + O\big(h^{n+1}\big), \tag{3}$$

which is the expression used in [39,38,35].

In this work we simplify the procedure for evaluating of $J_\alpha(x_{i+1})$ used in [39,38,35] by casting the jump correction as the result of a polynomial extrapolation. To do so we define the degree n interpolating polynomial $p_n(x) = \sum_j f(x_j)\ell_j(x)$

and note that $(f^{(k)})^{-}_{FD} = p_n^{(k)}(x_\alpha)$. Making this substitution in (3) reveals that $J_\alpha(x_{i+1})$ is an $(n+1)$ term Taylor expansion of $-p_n(x)$ about the point x_α, which is equivalent to the evaluation

$$J_\alpha(x_{i+1}) = -p_n(x_{i+1}). \tag{4}$$

Thus the evaluation of n one-sided finite differences can be replaced with a single evaluation of the interpolating polynomial via Neville's algorithm [40]. This implies that at domain boundaries the jump-corrected finite differences of the EJIIM are equivalent to a polynomial extrapolation of $f(x)$ followed by the evaluation of a standard finite difference stencil, which closely links our specialization of the EJIIM to other extrapolation-based procedures such as sharp immersed boundary methods [8,9] and the Ghost Fluid method [12].

This result generalizes to a variety of situations beyond the one described above. If $f(x)$ is nonzero for $x > x_\alpha$ due to an uncoupled physical process occurring on the other side of the interface, then the contributions to $J_\alpha(x_{i+1})$ from $(f^{(k)})^{+}$ can be approximated by an evaluation of $f(x)$:

$$J_\alpha(x_{i+1}) = f(x_{i+1}) - p_n(x_{i+1}) + O\left(h^{n+1}\right). \tag{5}$$

If a Neumann condition is present at x_α instead of a Dirichlet condition, then $p_n(x)$ is replaced by the unique interpolating polynomial satisfying $p^{(1)}(x_\alpha) = f^{(1)}(x_\alpha)$ and interpolating $f(x)$ at the n neighboring points $\{x_{i-1}, \ldots, x_{i-n}\}$. If no boundary condition is available, then the boundary point x_α is replaced by x_i in the interpolation stencil. All of these specializations of the EJIIM reduce complexity and computational cost. They also improve numerical stability, by eliminating the need to invert a Vandermonde matrix when calculating one-sided stencil coefficients. We view this simplification as an advantage over prior one-sided IIM formulations based on the EJIIM [34,35] and use this procedure continually in the remainder of this work.

2.2. Extending the IIM to 2D and nonconvex geometries

The sharp-interface method inspired by the EJIIM, as described above, extends readily to multidimensional problems, where the polynomial extrapolation perspective further helps address the challenges associated with concave geometry. To discuss multidimensional immersed interface methods, we define some convenient terminology and notation. Let $\mathcal{X} = \{ih,\ 0 \le i \le N_x\}$ and $\mathcal{Y} = \left\{jh,\ 0 \le j \le N_y\right\}$ represent the sets of x and y coordinate values for a uniform Cartesian grid, and let $\mathcal{G} = \mathcal{X} \times \mathcal{Y}$ be the set of all grid points. Further, let Ω be an irregular two-dimensional domain immersed in this grid. Following the lines of Gillis et al. [38], we define two additional sets of points:

- The set of **Control points** (denoted by C) contains all intersections between the grid lines of the Cartesian grid and the immersed boundary $\partial\Omega$. Specifically, $C = C_x \cup C_y$, where $C_x = \{(x, y) : (x, y) \in \partial\Omega \text{ and } x \in \mathcal{X}\}$ is the set of intersections between $\partial\Omega$ and grid lines with constant x coordinate, while $C_y = \{(x, y) : (x, y) \in \partial\Omega \text{ and } y \in \mathcal{Y}\}$ is the set of intersections between $\partial\Omega$ and grid lines with constant y coordinate.
- The set of **Affected points** (denoted by $\mathcal{A}$) are regular grid points that are adjacent to a control point. $\mathcal{A}$ can be subdivided into two disjoint sets $\mathcal{A}^+$ and $\mathcal{A}^-$, which represent points that are inside and outside Ω, respectively.

The sets C, $\mathcal{A}^-$, and $\mathcal{A}^+$ are labeled in Fig. 2 for a typical domain. Section S1 of the supplementary material provides a method for efficiently calculating the set of control points C for smooth geometries represented by a level set. With the control points found, our 2D immersed interface method precedes each finite-difference evaluation by a polynomial extrapolation which gives nonzero values to the inner affected points $\mathcal{A}^-$. For inner affected points with only one neighboring control point, this value is determined using a one-dimensional polynomial extrapolation along the associated grid line (as discussed in section 2.1). This is equivalent to the dimension-splitting techniques implemented in [38,35]. For inner affected points with multiple neighboring control points, we follow Marichal et al. [39] by averaging the results of the 1D extrapolations along associated grid lines. Both of these cases are illustrated in Fig. 2. The averaging procedure preserves the order of accuracy of the extension, and provides a minimal way to reconcile 1D extrapolations taken from different directions.

For nonconvex geometries, some control points may not have enough immediate neighbors to form an interpolating polynomial of the correct degree. This challenge does not disappear as the grid is refined: any amount of concavity, no matter how slight, can lead to control points with as few as one immediate neighbor (Fig. 2). One way to avoid this is to rely on extrapolations taken from multiple coordinate directions, as discussed above. This strategy is suggested and successfully implemented by Hosseinverdi and Fasel [32] for an immersed interface method based on compact finite differences. In the method presented here, control points with too few neighbors are simply ignored, and the associated points in $\mathcal{A}^-$ are filled using an extrapolation along a different coordinate direction. For smooth geometries, this extrapolation method allows every point in $\mathcal{A}^-$ to be filled given sufficient resolution. Nonsmooth geometries present additional challenges, including cusps and acute interior corners, which are left for future work. Overall this extension along multiple coordinate axes allows the IIM to operate on nonconvex geometries that could not be successfully discretized with previous one-sided IIM formulations [34,35], without requiring the use of complex multidimensional interpolants.

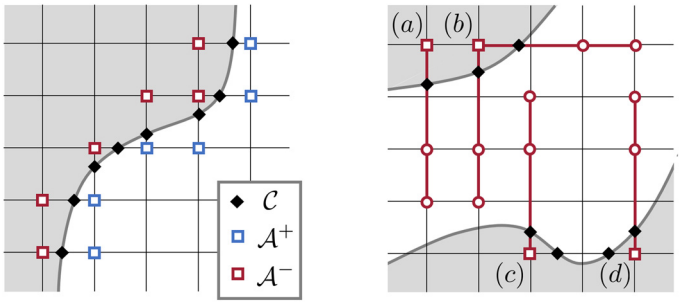


Fig. 2. (left) The control points $\mathcal{C}$ and affected points $\mathcal{A}$ for a typical IIM discretization. The gray region represents points lying outside the problem domain. (right) Stencils used to fill the affected points $\mathcal{A}^-$ with a third order extrapolation. Point (a) has only one neighbor in the problem domain, and consequently receives an extrapolation along only one coordinate direction. Point (b) has two neighbors, so its value is the average of two separate one-dimensional extrapolations. Points (c) and (d) have two neighbors in the problem domain, but there are not enough points in the domain to allow for a third-order horizontal extrapolation stencil; consequently, each is filled from a vertical extrapolation only.

3. Vorticity transport

We now proceed with combining our IIM with the discretization of transport equations in 1D and 2D. Specifically, we focus on the vorticity evolution equation governing the 2D incompressible Navier-Stokes equation, written here in conservative form:

$$\frac{\partial \omega}{\partial t} + \nabla \cdot (\mathbf{u}\omega - \nu \nabla \omega) = 0. \tag{6}$$

Integrating this differential conservation law over a 2D region R leads to the integral form

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_R \omega \, \mathrm{d}A + \oint_{\partial R} (\mathbf{u}\omega - \nu \nabla \omega) \cdot \hat{\mathbf{n}} \, \mathrm{d}s = 0. \tag{7}$$

In this section, we discretize (6) with a conservative finite difference scheme, using numerical fluxes as described in Shu [41]. While Lagrangian advection schemes are common in vorticity-based discretizations of the 2D Navier-Stokes equations [5,7,42,43], including other immersed interface methods [39,35], we have chosen a conservative Eulerian transport scheme because it offers an explicit notion of the flux of a conserved quantity through a surface. This leads to a discrete form of the integral conservation law (7), which is essential to the discretization of Kelvin's theorem presented in section 5.1. We also develop an immersed interface boundary treatment that respects the mixed hyperbolic-parabolic character of (6), and show that it does not degrade the stability or accuracy of the free-space scheme.

3.1. Free space discretization

For illustration, we begin with the one-dimensional advection-diffusion equation

$$\frac{\partial \omega}{\partial t} + \frac{\partial}{\partial x}\left(u\omega - \nu \frac{\partial \omega}{\partial x} \right) = 0, \tag{8}$$

with a spatially varying velocity u and constant diffusivity ν. Let $f = u\omega$ be the advective flux, and $q = -\nu\omega_x$ be the diffusive flux, so that $\omega_t + (f+q)_x = 0$. To discretize this equation, consider a one-dimensional grid of points x_i with regular spacing h, along with a discrete vorticity field ω_i and velocity field u_i. If numerical fluxes $f_{i+\frac{1}{2}}$ and $q_{i+\frac{1}{2}}$ are defined at the flux points $x_{i+\frac{1}{2}} = \frac{1}{2}(x_i + x_{i+1})$, then (8) can be approximated with centered differences:

$$\frac{\mathrm{d}\omega_i}{\mathrm{d}t} + \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{h} + \frac{q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}}{h} = 0. \tag{9}$$

The spatial accuracy of the conservative discretization depends on the interpolation procedure used to construct $f_{i+\frac{1}{2}}$ and $q_{i+\frac{1}{2}}$. Here we choose a third-order upwind advective flux using the stencils [41]

$$f_{i+\frac{1}{2}} = \begin{cases} -\frac{1}{6} f_{i-1} + \frac{5}{6} f_i + \frac{1}{3} f_{i+1}, & u_{i+\frac{1}{2}} \geq 0 \\ \frac{1}{3} f_i + \frac{5}{6} f_{i+1} - \frac{1}{6} f_{i+2}, & u_{i+\frac{1}{2}} < 0 \end{cases}, \tag{10}$$

where $f_i = u_i \omega_i$, and the local upwind direction is determined by $u_{i+\frac{1}{2}} = (u_i + u_{i+1})/2$. Compared to the standard second-order centered advective flux, the choice of a third order upwind scheme eliminates a dispersive truncation error of order

h^3 and produces a leading truncation error that is proportional to the fourth derivative of the flux. This hyper-diffusive truncation error provides a slight amount of numerical dissipation that is beneficial for the stability of the method. The diffusive flux $q_{i+\frac{1}{2}}$ is discretized with the centered difference

$$q_{i+\frac{1}{2}} = -\nu \frac{\omega_{i+1} - \omega_i}{h}, \tag{11}$$

which leads to overall second order accuracy for the diffusion term.

The 1D conservative transport discretization developed above is readily extended to two dimensions. For a 2D transport equation with velocity field $\mathbf{u} = (u_x, u_y)$, define the x-direction fluxes $f_x = u_x\omega$ and $q_x = -\nu\frac{\mathrm{d}\omega}{\mathrm{d}x}$ at the x-direction flux points $\mathbf{x}_{i+1/2,j} = (x_{i+1/2}, y_j)$, and the y-direction fluxes $f_y = u_y\omega$ and $q_y = -\nu\frac{\mathrm{d}\omega}{\mathrm{d}y}$ at the y-direction flux points $\mathbf{x}_{i,j+1/2} = (x_i, y_{j+1/2})$. Each of these fluxes is calculated by applying the one-dimensional schemes along the corresponding grid line, and the full transport equation is discretized using the differencing scheme

$$\frac{\mathrm{d}\omega_{i,j}}{\mathrm{d}t} + \frac{f_{i+1/2,j} - f_{i-1/2,j}}{h} + \frac{f_{i,j+1/2} - f_{i,j-1/2}}{h} + \frac{q_{i+1/2,j} - q_{i-1/2,j}}{h} + \frac{q_{i,j+1/2} - q_{i,j-1/2}}{h} = 0. \tag{12}$$

This 2D transport scheme obeys a discrete form of the integral conservation law (7). Because a similar discrete conservation property appears in the velocity reconstruction problem (section 4.2) and in the enforcement of Kelvin's theorem (section 5.1), we define notation for it here. Let R be a 2D rectangular region with boundaries passing through the flux points, enclosing the set of grid points $\{\mathbf{x}_{ij} : \ell_x \le i \le r_x,\ \ell_y \le j \le r_y\}$. The total vorticity in R can be approximated with the second-order quadrature

$$\int_R \omega \,\mathrm{d}A \approx h^2 \sum_{i=\ell_x}^{r_x} \sum_{j=\ell_y}^{r_y} \omega_{i,j} \equiv h^2 \sum_R \omega, \tag{13}$$

where we define the discrete operator $\sum_R(\cdot)$ as the sum over grid values within the region R. Similarly, a numerical flux $\mathbf{f}$ can be integrated over the boundary of R with the second-order discrete contour integral

$$\oint_{\partial R} \mathbf{f}\cdot\hat{\mathbf{n}}\,\mathrm{d}s \approx h\sum_{j=\ell_y}^{r_y} f_{r_x+1/2,j} + h\sum_{i=\ell_x}^{r_x} f_{i,r_y+1/2} - h\sum_{j=\ell_y}^{r_y} f_{\ell_x-1/2,j} - h\sum_{i=\ell_x}^{r_x} f_{i,\ell_y-1/2} \equiv h\sum_{\partial R} \mathbf{f}\cdot\hat{\mathbf{n}}, \tag{14}$$

where the four single summations represent the flux across the right, top, left, and bottom faces respectively, and $\sum_{\partial R}$ is shorthand for the sum over all these faces. Using this notation, the telescoping sum property of (12) leads to the discrete conservation law satisfied by our numerical scheme

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(h^2\sum_R \omega\right) + h\sum_{\partial R}(\mathbf{f}+\mathbf{q})\cdot\hat{\mathbf{n}} = 0, \tag{15}$$

which approximates the continuous integral conservation law to second order. This relation could be easily extended to more complex grid-aligned regions if needed, by noting that any such region can be written as a union of grid-aligned rectangles.

The stability of this free-space transport discretization is discussed in section S2 of the supplementary material; the scheme is conditionally stable when integrated with a second or third order Runge-Kutta scheme.

3.2. Immersed interface boundary treatment

For finite domains, the advection-diffusion equation (6) requires a single boundary condition for ω on each boundary. Here we consider the case where a Dirichlet boundary condition ω_b and the boundary velocity $\mathbf{u}_b$ are known in advance, which will be most relevant for a discretization of the full Navier-Stokes equations. Because of the distinct parabolic and hyperbolic nature of the diffusion and advection terms, the Dirichlet boundary condition is handled differently when discretizing each term.

We begin with the diffusive term. Let the grid points x_1 to x_N form a finite computational domain, with immersed boundaries at $x_\ell \in [x_0, x_1]$ and $x_r \in [x_N, x_{N+1}]$. To calculate the diffusive flux on a domain with immersed boundaries, the scalar field ω is extrapolated to x_0 and x_{N+1} using fourth-order polynomial extrapolations that depends on the Dirichlet boundary conditions ω_b, as described in section 2.1. Once this is done, the calculation of diffusive fluxes proceeds as usual for all flux points between $x_{1/2}$ and $x_{N+1/2}$. The use of a fourth-order extension leads to second-order accuracy for the diffusive term right up to the immersed boundary. This process is extended to 2D domains in a completely analogous way: the vorticity field is extrapolated to the inner affected points $\mathcal{A}^-$ at fourth order, using the Dirichlet boundary condition, and afterwards the diffusive fluxes are calculated normally.

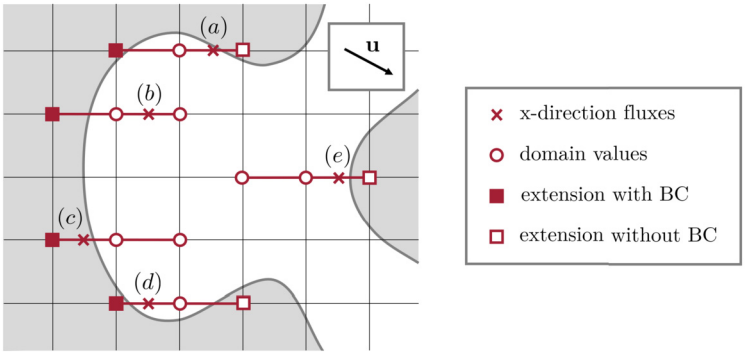


Fig. 3. Boundary treatment for the third-order upwind advective flux in 2D with a constant velocity field, as indicated in the upper right corner. Here the upwind direction for x-direction fluxes is to the left. Flux points with two neighbors in the domain are biased in the upwind direction (b), while fluxes with only one neighbor in the domain are biased towards that neighbor (a, c, d, e). For fluxes adjacent to an inflow boundary this amounts to the choice of a downwind stencil (c, d), while fluxes adjacent to an outflow boundary retain the upwind stencil (a, e). In all cases, whenever a stencil crosses the boundary, an extension with boundary condition is used on the upwind side (a, b, c, d), while an extension without boundary condition is used on the downwind side (a, d, e).

The calculation of the advection term requires a different procedure that addresses its hyperbolic nature. For a purely hyperbolic equation, boundary conditions are only necessary on regions of the boundary which act as an inflow. The use of a boundary condition on outflow boundaries leads to ill-posed continuous problems and instability in numerical discretizations. Consistent with this, we extrapolate the vorticity field past each domain boundary at third order, using the Dirichlet condition at inflow boundaries and ignoring the Dirichlet condition at outflow boundaries. The width of the third-order upwind advection stencil, which extends two points beyond each inflow boundary, presents an additional challenge. To avoid extra extrapolation, we force the choice of a downwind stencil at inflow boundaries, which extends only one point beyond the boundary. This strategy maintains the overall second-order accuracy of the transport discretization and does not affect the observed stability of the method.

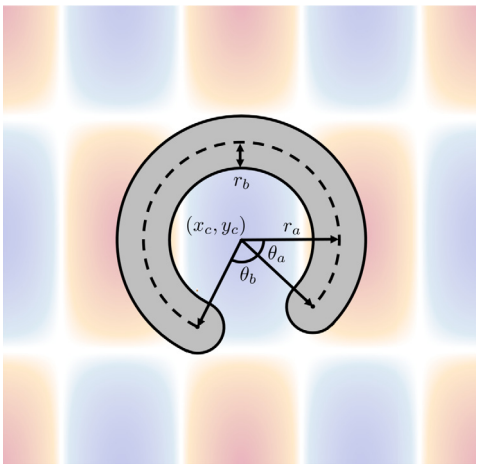
The 1D boundary treatment described above is readily extended to 2D domains. As in section 3.1, we continue to use a dimension split scheme, so that all advective fluxes are calculated using values from a single grid line and the upwind direction at each flux point is determined by the velocity component along that grid line. In two dimensions we require an advective flux value at each flux point which has at least one neighboring grid point in the fluid domain Ω. To obtain these values, the velocity field is extended to $\mathcal{A}^-$ at third order using the boundary velocity $\mathbf{u}_b$. This extended velocity field is used to determine the upwind direction at each flux point and to calculate the value of the advective flux at each grid point. From there on our treatment of the advective flux is determined by two separate policies, one governing the upwind or downwind bias of the third-order advection stencil and one governing the choice of boundary condition used in the extended vorticity field.

- **Stencil bias.** At flux points with two neighboring grid points in Ω, an upwind-biased stencil is used. At flux points with one neighboring grid point in Ω and one neighboring grid point in $\mathcal{A}^{-1}$, the stencil is biased away from the neighbor in $\mathcal{A}^-$.
- **Choice of vorticity extension.** The vorticity field is extended to $\mathcal{A}^-$ twice at third order, once with the Dirichlet condition ω_b and once without, so that both sets of values are available for flux calculations. All flux values along the grid line that are upwind of a given flux point are computed using the extension with boundary condition, while all flux values along the grid line that are downwind from the flux point are computed using the extension without boundary condition.

Fig. 3 illustrates the boundary stencils that are generated by this policy for several different inflow and outflow configurations, including for degenerate situations that arise with nonconvex geometries. Away from the domain boundaries, this policy coincides with the free space scheme described in section 3.1. Our choice of stencil bias alleviates the need to extend the vorticity field to additional points beyond $\mathcal{A}^-$, but reduces the accuracy of the advection scheme to second order at points adjacent to an inflow boundary. The choice of extension removes the influence of the Dirichlet boundary condition at outflow boundaries to respect the hyperbolic nature of the advection term.

3.3. Numerical results

To demonstrate that our immersed interface discretization of 2D the advection diffusion equation (6) is stable, second order accurate, and robust to concave geometry, we measure the convergence of our method on a 2D test case with a known analytical solution in both advection-dominant and diffusion-dominant regimes. A non-convex solid body is superimposed on the spatially periodic vorticity field



Element	Parameters
Grid	$\mathbf{x}_{ij} = (i/N,\ j/N)$ for $0 \leq i, j \leq N-1$.
Time steps	$T = 1.0, \delta t = 0.9\Delta t_{max}$
Solution	$\mathbf{u} = (1.0, 2.0)$, $\nu = (0.002, 0.064)$, $\mathbf{k} = (4\pi, 2\pi)$
Solid body	$\mathbf{x}_c = (0.507, 0.531)$, $r_a = 0.201$, $r_b = 0.054$, $\theta_a = 0.5$, $\theta_b = 2.4$

Fig. 4. Geometry and parameters used in the transport convergence study.

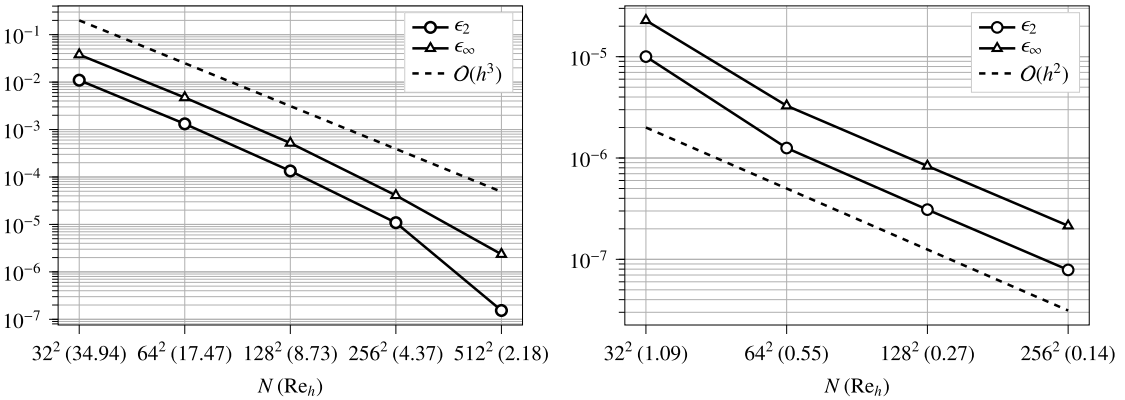


Fig. 5. Convergence behavior of the 2D transport scheme. L_2 and L_∞ error norms are plotted against the grid resolution, listed in terms of grid points N and cell Reynolds number $\mathrm{Re}_h = hU/\nu$. In advection dominant regimes with $Re_h > 1$ the spatial convergence is third order (left), falling to second order in diffusion dominant regimes with $\mathrm{Re}_h < 1$ (right).

$$\omega_{ex}(\mathbf{x}, t) = \cos[k_x(x - u_x t)]\cos[k_y(y - u_y t)]e^{-\nu k^2 t}, \tag{16}$$

which is an exact solution to the vorticity transport equation for constant velocity $\mathbf{u} = (u_x, u_y)$, viscosity ν, and wavenumber $\mathbf{k} = (k_x, k_y)$. The boundary conditions ω_b and $\mathbf{u}_b$ on the solid body are set to match this solution, and a periodic boundary condition is prescribed on the edge of the computational domain. The flow is discretized on a square grid with N points along each side, and integrated from $t = 0$ to $t = T$ using a third order Runge-Kutta method. The time step Δt is chosen to be $0.9\Delta t_{max}$, where Δt_{max} is the maximum stable time step for the transport scheme (as determined by the procedure in section S2 of the supplementary material). Fig. 4 defines the geometry of the solid body and lists the exact discretization parameters used in this test case.

The convergence of the numerical solution is measured with the L_2 and L_∞ error norms

$$\epsilon_2 = \sqrt{h^2 \sum_{\mathbf{x}_{ij} \notin B} \left(\omega_{ij}(T) - \omega_{ex}(\mathbf{x}_{ij}, T)\right)^2}, \tag{17}$$

$$\epsilon_\infty = \max_{\mathbf{x}_{ij} \notin B} \left(\omega_{ij}(T) - \omega_{ex}(\mathbf{x}_{ij}, T)\right). \tag{18}$$

Stability constraints require that the maximum time step Δt_{max} is $O(h)$ in advection-dominant regimes ($\mathrm{Re}_h = (|u_x| + |u_y|)h/\nu \gg 1$) and $O(h^2)$ in diffusion-dominant regimes ($\mathrm{Re}_h \ll 1$). Consequently, both the spatial and temporal truncation errors should be $O(h^3)$ in advection-dominant regimes, and the $O(h^2)$ spatial error should dominate in diffusion-dominant regimes. Fig. 5 plots the L_2 and L_∞ error norms against spatial resolution for parameter sets in both regimes, demonstrating the expected rate of convergence in each one.

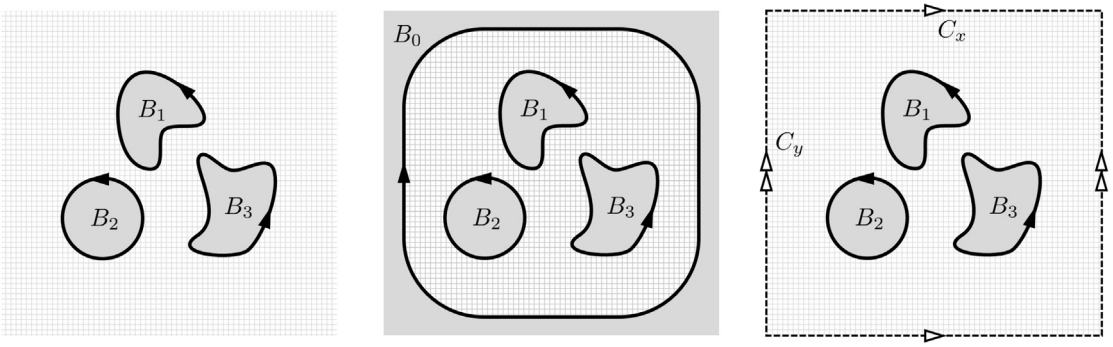


Fig. 6. Three exterior boundary conditions considered for the velocity reconstruction problem: an unbounded fluid domain (left), a bounded fluid domain with solid boundaries (center), and a periodic domain with the top/bottom and left/right boundaries identified (right). Arrows indicate the direction of the tangential unit vector along each boundary.

4. Velocity reconstruction

The vorticity-velocity formulation relies on a kinematic equivalence between the vorticity and velocity field. Obtaining the vorticity from the velocity is local and inexpensive; obtaining the velocity from the vorticity, as discussed here, requires the solution of an elliptic PDE. This reconstruction procedure has been widely discussed, but rarely with a focus on 2D domains with multiple immersed bodies. In this section we review the continuous theory, and discuss the conditions under which the velocity reconstruction problem has a unique solution in multiply connected domains. This continuous formulation is then discretized using a generalization of the immersed interface Poisson solver developed by Gillis et al. [38], and the resulting algorithm is shown to achieve second order accuracy in reconstructing a velocity field on a multiply connected domain.

4.1. Continuous formulation

Given a scalar vorticity field ω with compact support on a two-dimensional fluid domain Ω, the objective of the velocity reconstruction problem is to find a divergence-free velocity field $\mathbf{u}$ satisfying $\nabla \times \mathbf{u} = \omega$, along with no through-flow boundary conditions on solid bodies. Collecting all of these requirements yields the boundary value problem

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \quad \text{on} \quad \Omega, \\ \nabla \times \mathbf{u} &= \omega \quad \text{on} \quad \Omega, \\ \hat{\mathbf{n}} \cdot \mathbf{u} &= \hat{\mathbf{n}} \cdot \mathbf{u}_b \quad \text{on solid boundaries.} \end{aligned} \tag{19}$$

In general, this is not enough to specify a unique velocity field. The question of existence and uniqueness of solutions to (19) is deeply tied to the topology of Ω — see Cantarella et al. [44] for a complete exposition. The discussion here is limited to three cases that are common in 2D fluid simulations: unbounded domains, bounded domains, and periodic domains.

In 2D, it is simplest to analyze the reconstruction problem by re-writing the velocity field in terms of a stream function. For concreteness, assume that the fluid domain Ω is connected, that it contains distinct solid bodies B_k, $1 \leq k \leq N_b$, and that it is bounded in one of three ways: by a free-space boundary condition, by a solid exterior boundary B_0, or by a periodicity constraint (Fig. 6). On solid boundaries, let $\hat{\mathbf{n}}$ be a unit normal vector that points into the fluid, and define the unit tangential vector $\hat{\mathbf{s}}$ so that $\hat{\mathbf{n}} \times \hat{\mathbf{s}} = \hat{\mathbf{k}}$. A velocity field $\mathbf{u}(\mathbf{x})$ which solves (19) can be written in terms of a stream function whenever $\mathbf{u}_b$ satisfies

$$\oint_{\partial B_k} \mathbf{u}_b \cdot \mathbf{n} \, ds = 0 \quad \text{for } 0 \leq k \leq N_b. \tag{20}$$

This is the case whenever $\mathbf{u}_b$ is derived from the motion of solid bodies with constant area, and specifically holds throughout this work since we only consider stationary bodies and rotating cylinders. When (20) holds, let $\mathbf{u} = \nabla \times \psi$, so that $\nabla \cdot \mathbf{u} = 0$ automatically. Making this substitution in (19) yields a scalar Poisson equation for the stream function,

$$-\nabla^2 \psi = \omega \quad \text{on} \quad \Omega. \tag{21}$$

In terms of the stream function, the no through-flow boundary condition becomes $\partial_s \psi = \mathbf{u}_b \cdot \hat{\mathbf{n}}$. Under condition (20), $\partial_s \psi$ can be integrated around the boundary of each body to obtain a single-valued function

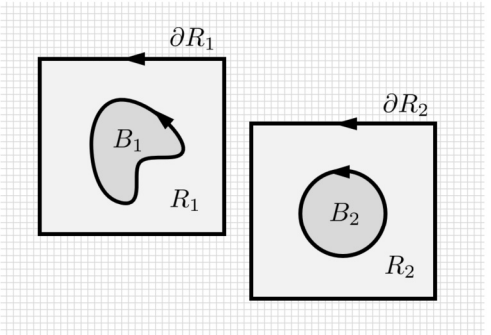


Fig. 7. The circulation around the boundary of a region R_k is directly related to the circulation on an enclosed object B_k, through the vorticity field and Stokes' theorem.

$$\psi_b(s) = \int_{s_0}^{s} \mathbf{u}_b \cdot \hat{\mathbf{n}} \, \mathrm{d}s \,. \tag{22}$$

The no through-flow boundary condition can then be expressed as the Dirichlet boundary condition

$$\psi = \psi_b + \bar{\psi}_k \quad \text{on } B_k, \tag{23}$$

where each $\bar{\psi}_k$ is an unknown constant associated with body B_k.

The presence of the unknown constants $\bar{\psi}_k$ indicates that (19) has multiple solutions: different choices of the constants $\bar{\psi}_k$ lead to different velocity fields, all of which are valid solutions. An effective way to single out a particular velocity field is to specify the circulation of the velocity around each solid body,

$$\oint_{\partial B_k} \mathbf{u} \cdot \mathrm{d}\mathbf{s} = -\oint_{\partial B_k} \partial_n \psi \, \mathrm{d}s = \Gamma_k \quad \text{for} \quad 1 \le k \le N_b. \tag{24}$$

This provides N_b scalar constraints on the stream function, which fix the N_b arbitrary constants in the boundary condition. The circulation on body B_k can be specified directly, or by specifying the circulation around the boundary of a region R_k that contains B_k (Fig. 7). The equivalence follows immediately from Stokes' theorem, since

$$\oint_{\partial R_k} \mathbf{u} \cdot \mathrm{d}\mathbf{s} = \oint_{\partial B_k} \mathbf{u} \cdot \mathrm{d}\mathbf{s} + \int_{R_k \setminus B_k} \omega \, \mathrm{d}A \,. \tag{25}$$

While this is enough to specify a unique velocity field, it still leaves some ambiguity in the stream function, which is only defined up to a global additive constant. Fixing this gauge degree of freedom, as well as enforcing an exterior boundary condition, can be handled by a method specific to each type of domain topology (Fig. 6):

- If Ω is a bounded domain, the gauge degree of freedom can be fixed by specifying a value for the arbitrary constant $\bar{\psi}_0$ on the exterior boundary B_0. Here this is fixed at $\bar{\psi}_0 = 0$, essentially removing this variable from the reconstruction problem and leaving a degree of freedom only on each of the N_b interior solid bodies.
- If Ω is an unbounded domain, the stream function can be split into a free stream component $\psi_\infty = \mathbf{u}_\infty \times \mathbf{x}$ and a perturbation $\tilde{\psi}$ satisfying $\lim_{|\mathbf{x}|\to\infty} \left|\nabla \tilde{\psi}\right| = 0$. In practice, the perturbation $\tilde{\psi}$ is calculated by a convolution between a source field and a Green's function, and the gauge degree of freedom is fixed by the choice of an arbitrary constant in the Green's function.
- If Ω is a rectangular domain periodic in both directions, the notion of a free stream velocity is replaced by specifying the average velocity on the horizontal periodic boundary C_x and vertical periodic boundary C_y:

$$\bar{u}_x = \frac{1}{|C_y|} \int_{C_y} \mathbf{u} \cdot \hat{\mathbf{n}} \, \mathrm{d}s \,, \quad \bar{u}_y = \frac{1}{|C_x|} \int_{C_x} \mathbf{u} \cdot \hat{\mathbf{n}} \, \mathrm{d}s \,. \tag{26}$$

  The stream function can then be split into a periodic component $\tilde{\psi}$ and a non-periodic free stream component $\psi_\infty = \bar{\mathbf{u}} \times \mathbf{x}$. The gauge degree of freedom can be fixed by specifying that $\tilde{\psi}$ has zero mean ($\int_\Omega \tilde{\psi} \, \mathrm{d}A = 0$), which is convenient for solution methods that involve a Fourier transform. Finally, in periodic domains the N_b circulation constraints must satisfy the solvability condition

$$\int_\Omega \omega \, \mathrm{d}A = -\sum_{i=1}^{N_b} \Gamma_k, \tag{27}$$

which follows immediately from Stokes theorem. For $N_b = 0$ this reduces to $\int_\Omega \omega \, \mathrm{d}A = 0$.

These topology-specific exterior boundary conditions and gauge conditions, together with equations (19), (23), and (24), fully specify the stream function, and allow a unique velocity field to be reconstructed from the vorticity field, velocity boundary conditions, and body circulations. Here we will assume that the circulations are known in advance, and focus on discretizing the resulting elliptic equation; section 5.1 will address the issue of determining the body circulations.

4.2. Immersed interface velocity reconstruction

Because the primary component of the velocity reconstruction problem is a scalar Poisson equation, this section closely follows the 3D unbounded IIM Poisson Solver developed by Gillis et al. [38] and applied to 2D exterior flows in Gillis et al. [35]. The variation presented here allows for concave objects, and includes a novel discretization of the circulation constraints which allows for problems with multiple immersed bodies.

Consider functions ψ and ω defined on a Cartesian grid. The discrete Laplacian operator ∇_h^2 is discretized with the standard second-order five-point finite difference stencil, so that

$$\nabla_h^2 \psi_{i,j} = \frac{1}{h^2}(\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1} - 4\psi_{i,j}). \tag{28}$$

In a rectangular computational domain, the discretized Poisson equation $-\nabla_h^2 \psi = \omega$ can be solved efficiently with Fast Fourier Transforms (FFTs) when subject to unbounded, symmetric, or periodic boundary conditions [45]. This FFT-based solution procedure will be denoted by $(\nabla_h^2)^{-1}$, so that $\psi = -(\nabla_h^2)^{-1}\omega$ satisfies the discretized Poisson equation with the desired boundary treatment on the edge of the computational domain. The development in this section is agnostic to the particular choice of boundary conditions or gauge-fixing, so long as the resulting Poisson problem is well-posed and the operator $(\nabla_h^2)^{-1}$ assigns a unique solution to each source field ω. Any set of boundary condition for the velocity reconstruction problem discussed in section 4.1 is also a valid set of boundary conditions for the operator $(\nabla_h^2)^{-1}$ when specialized to the case of no immersed bodies ($N_b = 0$).

To extend this methodology to domains with immersed boundaries, we will consider a scalar Poisson equation $-\nabla^2 \psi = \omega$ with the Dirichlet boundary condition $\psi_b(s)$ defined in (22) prescribed on solid boundaries. For now, we ignore the unknowns $\bar{\psi}_k$ which appear in the boundary condition for the full velocity reconstruction problem. We continue to view the solution ψ and source field ω as functions defined on the entire Cartesian grid, now with $\psi = 0$ and $\omega = 0$ prescribed on the interior of each solid body. Thus for points that are not adjacent to the boundary, $-\nabla_h^2 \psi = \omega$ continues to hold in both the solid and fluid domains. For points in $\mathcal{A}$ (the set of points adjacent to the solid boundary $\partial\Omega$), the five-point Laplacian stencil crosses the solid boundary, and the IIM must be used to account for this.

To proceed, we define some convenient notation. Consider the vector spaces

- V_G, the space of functions defined on the entire Cartesian grid $\mathcal{G}$;
- $V_A \subset V_G$, the subspace of functions defined on the affected points $\mathcal{A}$;
- V_C, the space of functions defined on the control points C, which occur at intersections between the grid lines and the solid boundary $\partial\Omega$.

It is also helpful to define the inclusion operator $E_A : V_A \to V_G$, which reinterprets a function with support on $\mathcal{A}$ as a function defined on all of $\mathcal{G}$ by assigning zero values to $\mathcal{G} \setminus \mathcal{A}$. In this framework, ψ and ω are elements of V_G, while the Dirichlet boundary condition ψ_b resides in V_C. Using this notation, a Poisson equation discretized with the IIM takes the form

$$-\nabla_h^2 \psi + E_A \gamma = \omega, \tag{29}$$

where $\gamma \in V_A$ represents corrections to the standard finite difference stencil on solid boundaries. These corrections come from the polynomial extrapolation procedure outlined in section 2.2, here a fourth order extrapolation that uses the Dirichlet condition ψ_b. Consequently, γ is a linear function of both the boundary condition ψ_b and the unknown solution ψ, and can be written as

$$\gamma = A\psi + B\psi_b. \tag{30}$$

Here $A : V_G \to V_A$ and $B : V_C \to V_A$ are known linear operators. Together equations (29) and (30) form a system of linear equations for the unknown solution ψ and the unknown IIM corrections γ. This system can be reduced via a Schur complement to a smaller system involving only γ,

$$\left(I_A + A(\nabla_h^2)^{-1}E_A\right)\gamma = -A(\nabla_h^2)^{-1}\omega + B\psi_b, \tag{31}$$

where I_A is the identity operator on V_A. The use of the solution operator $(\nabla_h^2)^{-1}$ leads to a dense linear system. However, because this solution operator can be applied efficiently using FFTs, (31) can be solved efficiently with an iterative method. After solving for γ, the full solution ψ is determined by

$$\psi = -(\nabla_h^2)^{-1}(\omega - E_A\gamma). \tag{32}$$

The accuracy and computational efficiency of this Schur complement approach has been demonstrated extensively [38, 35] and similar Schur-complement approaches have been used in other immersed interface Poisson solvers [30,46]. An analogous immersed boundary formulation for the 3D velocity reconstruction problem is developed in [47], which also explores the spectral characteristics of such systems.

To extend the above methodology to the full velocity reconstruction problem, the unknown constants $\bar{\psi}_k$ must be added as additional unknowns, and the circulation constraints must be discretized to determine their values. Instead of imposing these constraints directly on immersed solid bodies, we choose to follow (25) and specify the circulation around the boundary of a rectangular region R_k that encloses each solid body B_k. This avoids integration over any immersed surfaces, and allows us to take advantage of a conservation property inherent in the standard 5-point Laplacian. Defining the x-direction numerical flux $(\nabla_h\psi)_{i+\frac{1}{2},j} = (\psi_{i+1,j} - \psi_{i,j})/h$ and y-direction numerical flux $(\nabla_h\psi)_{i,j+\frac{1}{2}} = (\psi_{i,j+1} - \psi_{i,j})/h$, the operator ∇_h^2 can be written as a conservative finite difference:

$$\nabla_h^2\psi = \frac{(\nabla_h\psi)_{i+\frac{1}{2},j} - (\nabla_h\psi)_{i-\frac{1}{2},j}}{h} + \frac{(\nabla_h\psi)_{i,j+\frac{1}{2}} - (\nabla_h\psi)_{i,j-\frac{1}{2}}}{h}. \tag{33}$$

Summing (33) over the points in R_k yields the discrete Gauss' theorem

$$-h^2\sum_{R_k}\nabla_h^2\psi = -h\sum_{\partial R_k}(\nabla_h\psi)\cdot\hat{\mathbf{n}}. \tag{34}$$

The right hand side of this relation is a second order approximation of the circulation around ∂R_k, which allows us to write a discrete circulation constraint

$$-h\sum_{\partial R_k}(\nabla_h\psi)\cdot\hat{\mathbf{n}} = \Gamma_k. \tag{35}$$

Making the substitution $\nabla_h^2\psi = -\omega + E_A\gamma$ in (34) leads to equivalent constraint on the unknown IIM corrections γ,

$$h^2\sum_{R_k}\omega - h^2\sum_{R_k}E_A\gamma = \Gamma_k. \tag{36}$$

To incorporate this constraint into an immersed interface Poisson solver, consider the following notation:

- $\sum_{B_k} : V_A \to \mathbb{R}$ sums all function values at affected points associated with B_k.
- $\sum_{R_k} : V_G \to \mathbb{R}$ sums all function values which fall within a region R_k.
- $\mathbb{1}_k \in V_C$ is the vector with ones at control points associated with B_k and zeros otherwise.

Introducing the circulation constraints as additional equations and the boundary constants $\bar{\psi}_k$ as unknowns, the discretized reconstruction problem becomes

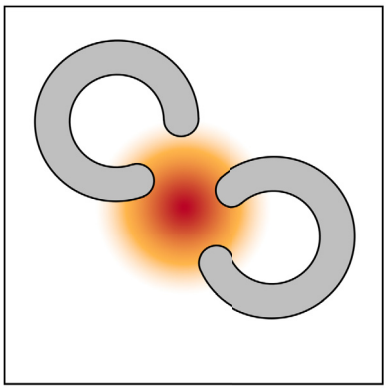
$$\left(I_A + A(\nabla_h^2)^{-1}E_A\right)\gamma - B\sum_{k=1}^{N_b}\bar{\psi}_k\mathbb{1}_k = -A(\nabla_h^2)^{-1}\omega + B\psi_b, \tag{37}$$

$$\Sigma_{B_k}\gamma = -\Gamma_{R_k} + \Sigma_{R_k}\omega \quad \text{for} \quad 1 \le k \le N_b. \tag{38}$$

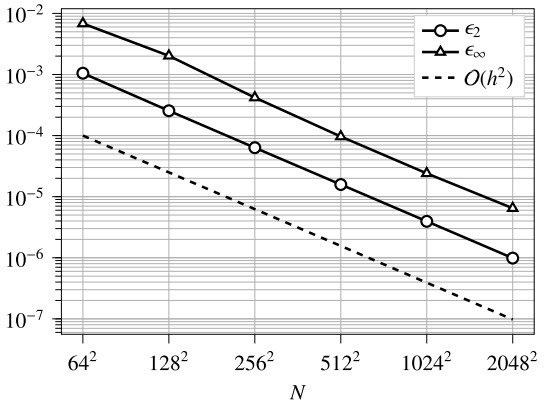
These equations are solved iteratively with the GMRes algorithm to find the values of the unknowns $(\gamma,\ \bar{\psi}_k)$, and the full solution ψ is recovered using (32).

Once the stream function has been determined, the velocity field $\mathbf{u} = \nabla\times\psi$ can be recovered. To do so in the presence of immersed bodies, the stream function is extrapolated beyond the solid boundaries using the boundary condition $\psi_b + \bar{\psi}_k$. The extrapolation used here is the same fourth-order procedure calculated by the operators A and B in the discretized reconstruction problem. The velocity at grid points within the domain is then calculated with the second-order centered difference

$$\mathbf{u}_{i,j} = \left(\frac{\psi_{i,j+1} - \psi_{i,j-1}}{2h}, -\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h}\right)^{\mathrm{T}}. \tag{39}$$



(a)



(b)

Element	Parameters
Grid	$\mathbf{x}_{ij} = (i/N,\ j/N)$ for $0 \le i, j \le N-1$.
Vortex	$\mathbf{x}_0 = (0.501, 0.501), \Gamma = 1.0, \nu t = 0.0025$
Body 1	$\mathbf{x}_c = (0.309, 0.679), r_a = 0.167, r_b = 0.057, \theta_a = 0.5, \theta_b = 1.8$
Body 2	$\mathbf{x}_c = (0.559, 0.451), r_a = 0.157, r_b = 0.057, \theta_a = 3.7, \theta_b = 2.0$

(c)

Fig. 8. (a and c) Geometry and parameter values used in the convergence study. (b) L_2 and L_∞ error norms for the velocity field as a function of grid resolution, showing second order convergence in both.

While a third-order extrapolation would be sufficient to maintain second order accuracy in the velocity field, we observed that the fourth-order extrapolation leads to a significantly reduced truncation error in the velocity field near solid boundaries, where velocity gradients tend to be the largest.

4.3. Numerical results

To demonstrate the accuracy and flexibility of our velocity reconstruction procedure, we consider a test case with non-convex solid bodies, a multiply connected domain, and free-space boundary conditions (Fig. 8a). Two bodies with geometry described in Fig. 4 are superimposed on the vorticity field of a Lamb-Oseen vortex,

$$\omega_{LO}(\mathbf{x}) = \frac{\Gamma}{4\pi\nu t}\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_0\|^2}{4\nu t}\right), \tag{40}$$

and the velocity $\mathbf{u}_b$ on their boundaries is prescribed to match the corresponding velocity field

$$\mathbf{u}_{LO}(\mathbf{x}) = \frac{\Gamma}{2\pi r^2}\left(1-\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_0\|^2}{4\nu t}\right)\right)\hat{\mathbf{k}}\times(\mathbf{x}-\mathbf{x}_0). \tag{41}$$

The reconstruction procedure described in the previous section is then used to recover the full velocity field, which is compared to the exact solution $\mathbf{u}_{LO}(\mathbf{x})$. This is done on a square Cartesian grid with N points along each side. To prescribe the circulations on solid bodies, each body is enclosed in a bounding region R_k, and the circulation on ∂R_k is estimated to second order from the original vorticity field:

$$\Gamma_{R_k} = h^2 \sum_{\mathbf{x}_{ij}\in R_k} \omega_{LO}(\mathbf{x}_{ij}). \tag{42}$$

The full details of the geometry, flow parameters, and bounding regions are provided in Fig. 8c. Fig. 8b plots the L_2 and L_∞ norms of the velocity error $\|\mathbf{u}-\mathbf{u}_{exact}\|$ against the spatial resolution N, demonstrating second order convergence in both error norms.

5. Navier-Stokes

In this section we introduce the components of our full Navier-Stokes discretization that do not fit neatly within the vorticity transport or velocity reconstruction problems. This includes a method for enforcing Kelvin's theorem, which is key

to simulations with multiple bodies; an outflow boundary condition for external flows; the vorticity boundary condition for the transport equation; and the calculation of forces and surface tractions. These are then combined with the transport scheme developed in section 3 and reconstruction scheme developed in section 4 to create an algorithm that solves the full Navier-Stokes equations.

5.1. Circulation in multiply connected domains

Section 4 outlined an algorithm that reconstructs the velocity field from the vorticity field, provided that the circulation around each solid body is known in advance. Continuously, the circulation around any body satisfying a no-slip condition can be determined by integrating the tangential component of the prescribed boundary velocity. However, for numerical algorithms that enforces the no-slip condition only approximately, including ours, a different strategy is needed to specify these circulations. Here we outline a method based on the enforcement of a discrete form of Kelvin's theorem.

For a 2D viscous flow, Kelvin's theorem states that the circulation around any material contour $C(t)$ evolves according to

$$\frac{\mathrm{d}}{\mathrm{d}t}\oint_{C(t)} \mathbf{u}\cdot \mathrm{d}\mathbf{s} = -\nu \oint_{C(t)} \frac{\partial \omega}{\partial n}\,\mathrm{d}s\,. \tag{43}$$

If C is a stationary contour, then an application of Reynolds transport theorem gives the equivalent expression

$$\frac{\mathrm{d}}{\mathrm{d}t}\oint_{C} \mathbf{u}\cdot \mathrm{d}\mathbf{s} = -\oint_{C} (\mathbf{u}\omega - \nu\nabla\omega)\cdot\hat{\mathbf{n}}\,\mathrm{d}s\,, \tag{44}$$

which is an ordinary differential equation that governs the evolution of the circulation around C. If C is the boundary of a simply connected fluid region R, then (44) can be derived directly from the vorticity transport equation and Stokes' theorem. However, if C is not the boundary of a simply connected fluid region, then Kelvin's theorem is a separate constraint from the vorticity transport equation that ensures the existence of single-valued pressure field [48].

For immersed interface methods, enforcing Kelvin's theorem has been challenging. In [34], Marichal observes that prescribing $\Gamma_{\partial B} = \Sigma_B \gamma = 0$ on a stationary solid body during velocity reconstruction leads to an unstable numerical scheme. The total circulation of the flow — as measured by the sum of all vorticity values on the grid — increases rapidly with time. The author avoids this instability by instead prescribing zero far-field circulation, through the condition

$$\Gamma_{\partial B} + h^2 \sum_{R_\infty} \omega = 0. \tag{45}$$

Here the vorticity field is defined to be zero inside solid bodies, R_∞ is a rectangular grid-aligned region containing the full support of the vorticity field, and we use the summation notation defined in equation (13). This condition directly enforces Kelvin's theorem by linking the circulation around a solid boundary to the vorticity created on that boundary. However, it does not generalize to flows in which vorticity is allowed to leave the computational domain at outflow boundaries, and it can only uniquely determine the circulation of one immersed body.

To generalize this condition, we note here that when both the vorticity transport equation and the velocity reconstruction problem are discretized with conservative finite differences, as we have done in sections 3 and 4, a discrete form of Kelvin's theorem holds automatically on the boundary of any rectangular grid-aligned fluid region. For any such region R, combining the conservation property of the transport equation (15) with the discrete Stokes theorem enforced by the reconstruction procedure (34) leads to the expression

$$-h\frac{\mathrm{d}}{\mathrm{d}t}\sum_{\partial R_k} (\nabla_h \psi)\cdot\hat{\mathbf{n}} = -h\sum_{\partial R} (\mathbf{u}\omega - \nu\nabla_h\omega)\cdot\hat{\mathbf{n}}, \tag{46}$$

which is analogous to (44). This discrete form of Kelvin's theorem immediately extends to any region which can be built by adding or removing a series of grid-aligned rectangles, each satisfying (46). If the region R is not purely fluid, but encloses a single immersed body B, then (46) no longer holds automatically. To remedy this, we begin by selecting R as the bounding region enclosing B in the velocity reconstruction procedure. We then treat Kelvin's theorem as an ODE which defines the evolution of the circulation Γ_R associated with R:

$$\frac{\mathrm{d}\Gamma_R}{\mathrm{d}t} = -h\sum_{\partial R} (\mathbf{u}\omega - \nu\nabla_h\omega)\cdot\hat{\mathbf{n}}. \tag{47}$$

This ODE is integrated with the same Runge-Kutta method as used in the transport equation, so that (46) holds automatically on ∂R. Condition (47) also enforces a discrete Kelvin's theorem on the boundary of every other grid-aligned region enclosing B, since any such region can be built from R by adding or removing rectangular fluid regions satisfying (46). Thus any choice of the region R on which we enforce (47) is equivalent to any other, and the dynamics of the discretized fluid system are independent of our choice of bounding regions.

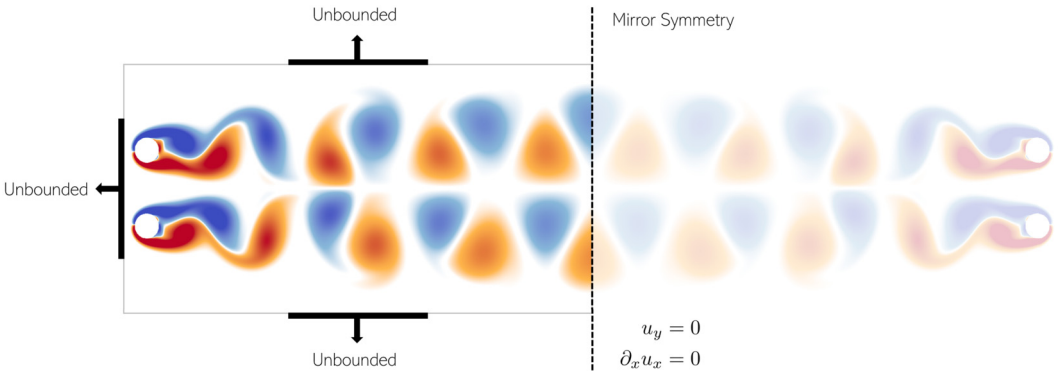


Fig. 9. Outflow boundaries are approximated by an even symmetry condition for the vorticity field on the outflow plane.

Taking R to be a contour R_∞ which encloses the entire vorticity field, we immediately find that condition (47) is equivalent to the far-field condition (45) proposed by Marichal for flows with a single immersed body. However, condition (47) does not require a contour which encloses the entire vorticity field, allowing for simulations with outflow boundary conditions. It can be also generalized to flows with multiple bodies by choosing a separate bounding region R_k for each body B_k. Here again, the specific choice of each bounding region R_k does not affect the discrete dynamics.

5.2. Outflow boundaries

The previous section introduced a circulation tracking scheme for vorticity-based immersed interface methods that does not require the computational domain to contain the entire vorticity field. This allows for the use of outflow boundary conditions, which are essential for long-time simulations of external flows. The outflow condition used here is a 2D analogue of the condition used for simulations of 3D wake dynamics in [49]. For a purely horizontal free stream, a vertical outflow plane is specified downstream of all immersed bodies, and the vorticity field is mirrored across this outflow plane as shown in Fig. 9. This leads to a reconstructed velocity field that satisfies

$$\begin{aligned} \frac{\partial u_x}{\partial x} &= 0, \\ u_y &= 0 \end{aligned} \tag{48}$$

on the outflow plane. The portion of the mirrored vorticity field adjacent to the outflow plane is used when calculating the vorticity flux in the vorticity transport equation. The rest of this mirror distribution is never constructed explicitly; it enters the velocity reconstruction problem through the use of a symmetry boundary condition in the operator $(\nabla_h^2)^{-1}$, as described by Caprace et al. [45].

5.3. Vorticity boundary conditions and vorticity flux

The vorticity-velocity form of the Navier-Stokes equations requires a no-slip velocity boundary condition on immersed surfaces. However, because the velocity field is reconstructed from the vorticity field through an elliptic equation, it is difficult to translate the no-slip velocity boundary condition into a boundary condition for the vorticity transport equation. The approach taken here is a minor variation on the method used by Gillis et al. [35], which allows for high order explicit time integration and nonconvex immersed bodies. It is similar in spirit to the immersed interface vorticity boundary condition used by Linnick and Fasel [31], and falls into the class of local vorticity boundary conditions originated by Thom [50] and catalogued by E and Liu [51]. Other notable strategies are the vorticity integral constraints developed by Quartapelle [52], and the Lighthill splitting approach (which was investigated extensively in the context of immersed interface methods by Marichal [34]). However, neither the integral constraints nor Lighthill splitting allow for the use of a high order explicit time integration scheme; Quartapelle's integral constraints require implicit integration, while Lighthill's splitting method is limited to first order temporal accuracy.

In the current method, the velocity reconstruction process yields a velocity field $\mathbf{u}$ that is defined on the Cartesian grid points. The velocity field at control points $\mathbf{x}_c \in C$ is known from the velocity boundary condition $\mathbf{u}_b$. Using this information, both components of the velocity field are extended past the domain boundary using a third order polynomial extrapolation. The boundary vorticity ω_b is taken to be the curl of the extended velocity field evaluated on the boundary, and is evaluated using the second-order stencils shown in Fig. 10. This is a more compact scheme than the stencils used by Gillis et al. [35], intended to better accommodate concave geometries.

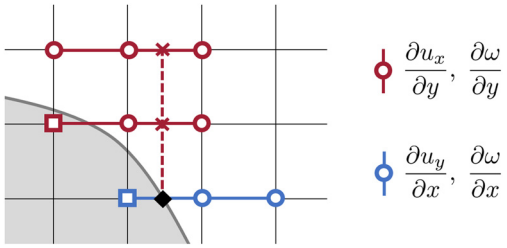


Fig. 10. Stencils used to calculate the vorticity boundary condition at second order. The blue points define a three-point stencil for the x-direction velocity derivative on the boundary, which uses three neighboring velocity values defined on the Cartesian grid. The y-direction velocity derivative is more involved; the value of u_x at the red crosses is interpolated using three-point stencils, and these values are then used along with the velocity boundary condition to calculate the y-direction velocity derivative at the boundary. This same set of stencils is also used to calculate the boundary vorticity gradient. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

5.4. Force calculations

The total lift force and drag force acting on an immersed body can be calculated using the control volume formulations derived by Noca [53], and the total torque acting on the body can be calculated using an analogous control volume formulation which we derive in sections S3 and S5 of the supplementary material. The shear stress distribution on a stationary immersed surface can be derived directly from the surface vorticity: $\tau(s) = \nu\omega(s)$, where s is a surface coordinate.

Recovering the surface pressure distribution is more involved, and relies on the relation between the surface pressure gradient and surface vorticity gradient. On stationary solid boundaries with a no-slip condition, as considered here, this is

$$0 = -\nabla p - \nu \nabla \times \omega. \tag{49}$$

To evaluate the vorticity gradient, the vorticity field is extended past the domain boundary using a third order extrapolation, taking into account the vorticity boundary condition ω_b computed as in section 5.3. The derivative of this extended field along the coordinate directions is calculated using the same stencils points as the vorticity boundary condition (Fig. 10), and then projected onto the local normal and tangential unit vectors. With this gradient known, we can consider two distinct methods of pressure recovery. In the first, the tangential pressure gradient is calculated from the normal vorticity flux, and then integrated over the immersed surface. Beginning the integration at an arbitrary point with surface coordinate s_0 leads to the expression

$$p(s) - p(s_0) = \int_{s_0}^{s} \nu \frac{\partial \omega}{\partial n} \, \mathrm{d}s \,. \tag{50}$$

Because the fluid adjacent to the boundary forms a material contour, Kelvin's theorem (43) guarantees that this integral is single-valued. The additive constant $p(s_0)$ cannot be determined from boundary information alone, but this procedure is still useful for measuring relative pressure differences over the immersed surface. To avoid the unknown additive constant $p(s_0)$, we also consider a second and more expensive method of pressure recovery. Following Lee et al. [54], we define the total pressure

$$H = (p - p_\infty) + \frac{1}{2}\left(|\mathbf{u}|^2 - |\mathbf{u}_\infty|^2\right), \tag{51}$$

which satisfies the far-field condition $H \to 0$ as $x \to \infty$ and the scalar Poisson equation

$$\nabla^2 H = \nabla \cdot (\mathbf{u} \times \omega \hat{\mathbf{k}}). \tag{52}$$

A Neumann boundary condition for the equation can be constructed from the definition of H and the normal component of (49):

$$\left.\frac{\partial H}{\partial n}\right|_S = \frac{\partial p}{\partial n} - \frac{\partial}{\partial n}\left(\frac{|\mathbf{u}|^2}{2}\right) = -\nu \frac{\partial \omega}{\partial s} - \frac{\partial}{\partial n}\left(\frac{|\mathbf{u}|^2}{2}\right). \tag{53}$$

Discretely, the tangential vorticity gradient and the normal gradient of $|\mathbf{u}|^2/2$ are calculated using the boundary stencils described in section 5.3. The pressure Poisson equation (52) is then solved using the IIM Poisson solver outlined in section 4.2. To handle the Neumann boundary condition, we use the compatible extrapolation procedure developed by Marichal et al. [39]. The resulting pressure field is interpolated back to the immersed solid boundaries using a third order polynomial extrapolation to obtain the surface pressure $p(s)$.

5.5. Full algorithm for flow simulations

Having established a transport scheme, a velocity reconstruction scheme, a method for enforcing Kelvin's theorem, and a vorticity boundary condition, we can lay out a complete algorithm for solving the 2D incompressible Navier-Stokes equations in vorticity form. The dynamic variables are the discretized vorticity field $\omega(\mathbf{x}, t)$ and the N_b bounding box circulations $\Gamma_k(t)$, both of which require an initial conditions $\omega_0(\mathbf{x})$ and $\{\Gamma_{k,0}\}$. These can be specified directly, or inferred from an initial velocity field $\mathbf{u}_0(\mathbf{x})$. Together the discretized vorticity field and circulation are determined by a large system of ODEs,

$$\frac{\partial \omega}{\partial t}, \left\{\frac{\mathrm{d}\Gamma_k}{\mathrm{d}t}\right\} = f(\omega, \{\Gamma_k\}), \tag{54}$$

defined by the following sequence:

- **Velocity Reconstruction.** Given the vorticity field ω, circulations $\{\Gamma_k\}$, and velocity boundary condition $\mathbf{u}_b$, the stream function ψ is recovered by solving the scalar Poisson equation

$$\begin{aligned} -\nabla^2 \psi &= \omega \quad \text{on} \quad \Omega, \\ \psi &= \psi_b + \bar{\psi}_k \quad \text{on} \quad \partial B_k, \\ -\oint_{R_k} \partial_n \psi \, \mathrm{d}s &= \Gamma_k \quad \text{for} \quad 1 \leq k \leq N_b. \end{aligned}$$

  This is done with the reconstruction procedure outlined in section 4. The velocity field $\mathbf{u}$ is calculated by differentiating the stream function.
- **Vorticity Transport.** The Dirichlet boundary condition ω_b for the vorticity transport equation is calculated using the local method outlined in section 5.3. The transport equation

$$\frac{\partial \omega}{\partial t} = -\nabla \cdot (\mathbf{u}\omega - \nu \nabla \omega)$$

  then determines the time derivative of the vorticity field, through the conservative spatial discretization developed in section 3.
- **Kelvin's Theorem.** As outlined in section 5.1, the time derivative of the circulations is determined by Kelvin's theorem,

$$\frac{\mathrm{d}\Gamma_k}{\mathrm{d}t} = -\oint_{\partial R_k} (\mathbf{u}\omega - \nu \nabla \omega).$$

  The spatial integration is performed with the same vorticity flux used in the transport scheme.

This system of ODEs is integrated in time with a low storage third-order Runge-Kutta scheme [55]. The size of each time step is chosen to be a fixed fraction $0 < C_{\text{stab}} < 1$ of the maximum stable time step for the transport scheme, calculated using the procedure outlined in section S2 of the supplementary material. All force and pressure calculations are performed after the first velocity reconstruction of each time step, when the vorticity and circulations have third-order temporal accuracy.

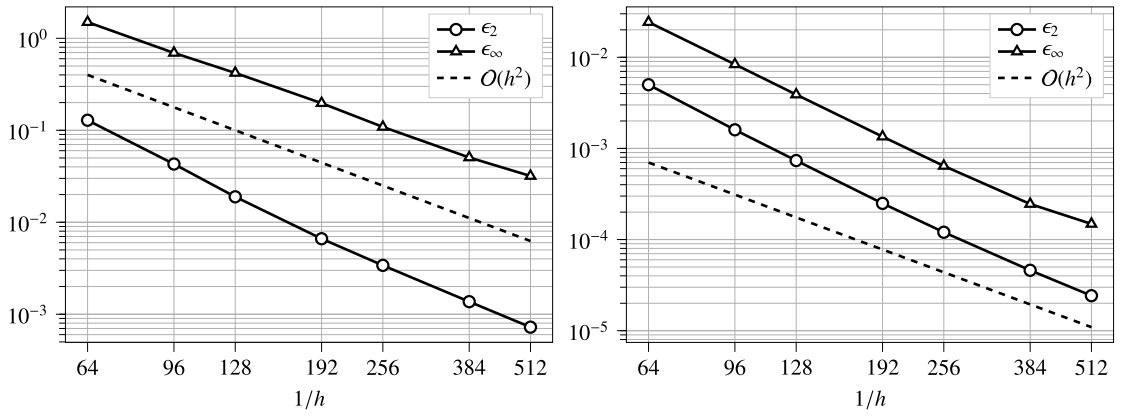
The algorithm described above has been implemented in C++ using Cubism [56], a library for block-based parallelism on uniform resolution Cartesian grids. The velocity reconstruction problem is solved with FFT-accelerated convolutions performed by FLUPS [45], which allows for fast solutions of scalar Poisson equations on rectangular domains with arbitrary combinations of unbounded, symmetric, and periodic boundary conditions.

6. Results

The Navier-Stokes discretization developed in the previous sections is applicable to a broad class of 2D incompressible flows. Here we first demonstrate the convergence of the method for a simple test case with an analytical solution, and then illustrate the effectiveness of this discretization in calculating vorticity fields, velocity fields, and surface traction distributions for a variety of external and internal flows.

6.1. Convergence: Lamb-Oseen vortex

To demonstrate the convergence of the 2D Navier-Stokes discretization developed here, we consider an external flow test case with a solid body and an analytical solution, as done in Gillis et al. [35]. A rotating cylinder with radius R and center $\mathbf{x}_c$ is immersed in a uniform Cartesian grid with grid spacing h. The initial vorticity field outside of cylinder is set to match the vorticity field of a Lamb-Oseen vortex centered at $\mathbf{x}_c$, and the cylinder's time-dependent rotation rate is set so that at



(a) Vorticity error.

(b) Velocity Error

Element	Parameters
Grid	$\mathbf{x}_{ij} = (i/N,\ j/N)$ for $0 \le i, j \le N-1$.
Time	RK3 with safety factor $C_{stab} = 0.7$. $t_0 = 1.0$, $t_f = 2.0$.
Cylinder	$\mathbf{x}_c = (0.507, 0.507)$, $R = 0.15$
Vortex	$\Gamma = \pi$, $\nu = 0.001$

(c) Parameters used in the full Navier-Stokes convergence study.

Fig. 11. Convergence of the error in the (a) vorticity field and (b) velocity field with spatial resolution for the full Navier-Stokes discretization, as well as the parameter values used in the convergence study (c).

each point on the solid boundary $\mathbf{u}_b(t) = \mathbf{u}_{LO}(R, t)$. This ensures that the Lamb-Oseen vortex is an analytical solution to the flow outside of the cylinder which satisfies the no-slip boundary condition for all time. The flow is integrated from time t_0 to time t_f using a third-order Runge-Kutta scheme, and the numerical vorticity and velocity fields are compared to the analytical vorticity and velocity fields using the L_2 and L_∞ error norms defined in sections 3.3 and 4.3. The full details of the grid, cylinder, vortex, and time integration are provided in Fig. 11c.

Fig. 11 plots the L_2 and L_∞ error norms of the velocity and vorticity fields against the spatial resolution h, demonstrating second order convergence in both norms for both fields. We emphasize that the convergence rate of the L_∞ error norms indicates that the full Navier-Stokes algorithm achieves second-order spatial accuracy right up to the immersed boundary.

6.2. Impulsively rotated cylinder

The flow around an impulsively rotated cylinder is another viscous exterior flow with an analytical solution, and an excellent test case for the enforcement of Kelvin's theorem. Consider a cylinder of radius R immersed in a quiescent fluid with viscosity ν, which begins rotating with constant angular velocity Ω at time $t = 0$. The impulsive start releases a singular vortex sheet into the flow, which then diffuses radially outward. Any mismatch between the magnitude of this thin vortex sheet and the rotation rate of the object leads to a violation of Kelvin's theorem, and can cause significant errors in the vorticity field and the resulting viscous moment acting on the cylinder.

For simplicity, the flow around the cylinder is assumed to remain axisymmetric. Using a non-dimensional time $t^* = \nu t/R^2$ and radial coordinate $r^* = r/R$, the non-dimensional vorticity distribution $\omega^* = \omega/\Omega$ is given by

$$\omega^*(r^*, t^*) = -\frac{2}{\pi}\int_0^\infty \Re\left\{\frac{K_0(ir^*x)}{K_1(ix)}\right\} e^{-x^2 t^*}\,\mathrm{d}x\,, \tag{55}$$

while the non-dimensional velocity distribution $u^* = u_\theta/R\Omega$ is given by

$$u^*(r^*, t^*) = \frac{1}{r^*} - \frac{2}{\pi}\int_0^\infty \Im\left\{\frac{K_1(ir^*x)}{K_1(ix)}\right\} \frac{e^{-x^2 t^*}}{x}\,\mathrm{d}x\,. \tag{56}$$

Here $K_0(x)$ and $K_1(x)$ are modified Bessel functions of the second kind, while $\Re$ and $\Im$ denote the real and imaginary parts respectively. These analytical expressions are derived in section S4 of the supplementary material and provide a more easily evaluated result for the velocity field compared to the expressions provided by Lagerstrom [57]. The integrands in (55) and (56) are non-singular at $x = 0$ and decay rapidly as $x \to \infty$, so that the improper integrals can be evaluated numerically with

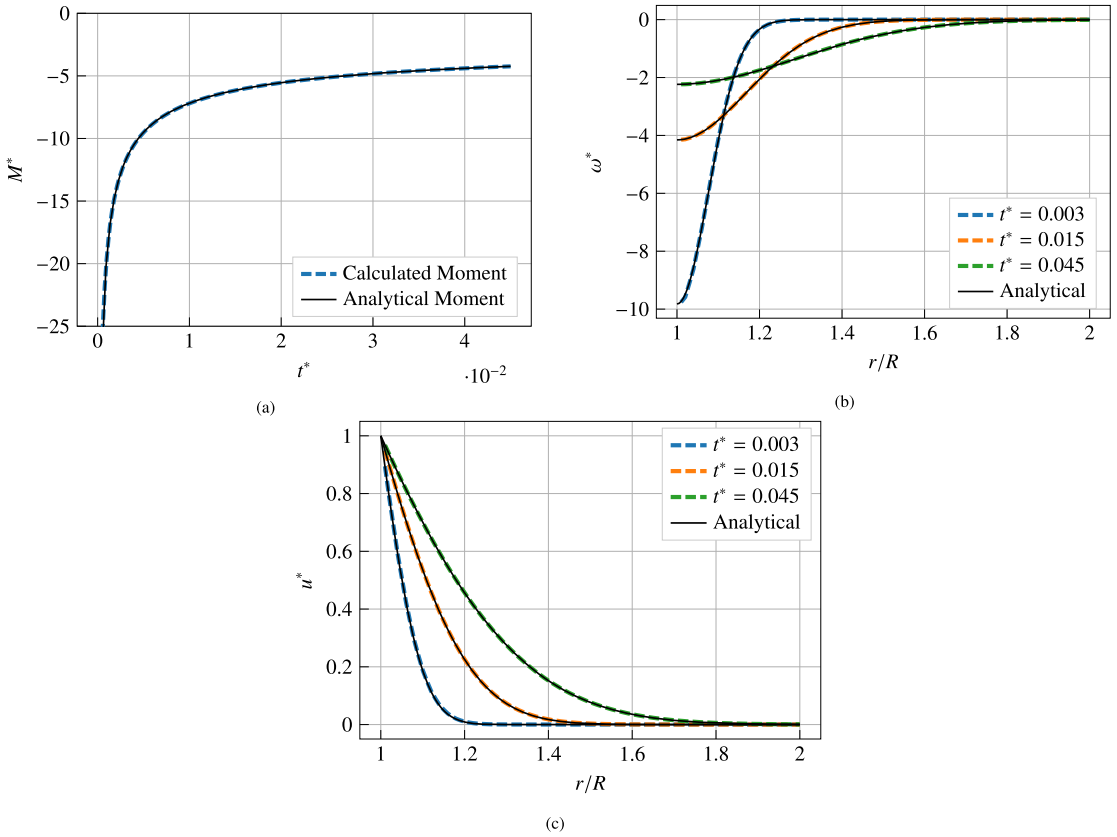


Fig. 12. Numerical and semi-analytical results for the impulsively rotated cylinder, including (a) the time history of the total moment, (b) the vorticity field at selected times, and (c) the velocity field at selected times.

good accuracy. The total non-dimensional moment $M^* = M/2\pi R^2 \nu\Omega$ acting on the cylinder can be calculated by integrating the resulting shear stress distribution, giving $M^*(t^*) = \omega^*(1, t^*) - 2$.

These analytical results are independent of the Reynolds number $\mathrm{Re}_\Omega = R^2\Omega/\nu$. For the simulations discussed here, we have chosen $\mathrm{Re}_\Omega = 50$ to avoid any high Reynolds number instabilities that may disrupt the axisymmetric flow. Fig. 12 shows the time evolution of the moment acting on the impulsively rotated cylinder for $t^* \leq 0.045$, along with velocity and vorticity profiles at selected times. At the resolution shown here ($D/h = 104.2$), the numerical and semi-analytical results are in excellent agreement.

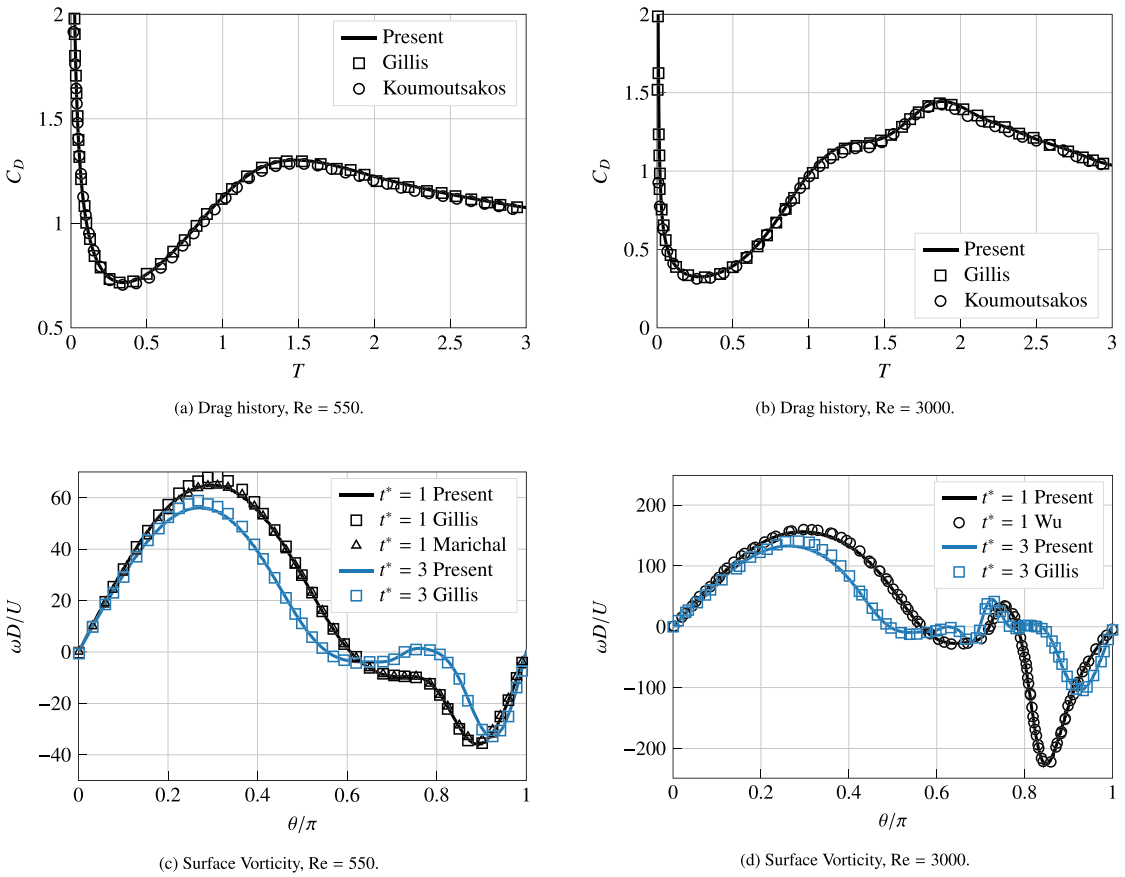
6.3. Impulsively translated cylinder

The impulsively translated cylinder is a widely used test case in two-dimensional incompressible flow [42,54,58,35,59]. Consider a cylinder of diameter D and center $\mathbf{x}_c = (x_c,\ y_c)$ immersed in an unbounded fluid with kinematic viscosity ν. At time $t = 0$, the cylinder begins translating with constant velocity, which produces a free-stream velocity of $\mathbf{u}_\infty = (u_{\infty,x},\ u_{\infty,y})$ in a reference frame attached to the cylinder. The dynamics of the resulting flow depend only on the Reynolds number $\mathrm{Re}_D = u_\infty D/\nu$. This section focuses on the short-time evolution of the flow-field, which takes place before the symmetry of the problem is broken and the commonly-observed vortex shedding behavior begins. In this symmetric regime there is no lift and no net moment acting on the cylinder, so that the drag force is the only relevant load.

Following Gillis et al. [35], the quality of the spatial discretization is measured with the two parameters

$$N_\delta = \frac{1}{\sqrt{\mathrm{Re}_D}}\left(\frac{D}{h}\right), \qquad Q = \mathrm{Re}_D^{3/2}\left(\frac{h}{D}\right)^2 = \frac{\mathrm{Re}_D^{1/2}}{N_\delta^2}. \tag{57}$$

The parameter N_δ estimates the number of grid points contained within the characteristic boundary layer thickness, while the parameter Q estimates the mesh Reynold's number based on the boundary vorticity. Gillis' results indicate that $N_\delta \sim 8$ or higher represents a well-resolved flow-field and is generally sufficient for obtaining accurate drag forces, while $Q \sim 1$ or lower allows for accurate wall vorticity values. Fig. 13 plots the drag coefficient $C_D = 2F/u_\infty^2 D$ as a function of the non-dimensional time $t^* = u_\infty t/D$ at two Reynolds numbers, $\mathrm{Re} = 550$ and $\mathrm{Re} = 3000$, for spatial resolutions $N_\delta = 8.73$ ($Q = 0.31$) and $N_\delta = 7.48$ ($Q = 0.98$) respectively. The drag coefficients are calculated using a control volume approach,



(a) Drag history, Re = 550.

(b) Drag history, Re = 3000.

(c) Surface Vorticity, Re = 550.

(d) Surface Vorticity, Re = 3000.

Fig. 13. Drag history and surface vorticity profiles for the impulsively started cylinder at Re = 550, $N_\delta = 8.73$, $Q = 0.31$ (left column) and $Re = 3000$, $N_\delta = 7.48$, $Q = 0.98$ (right column).

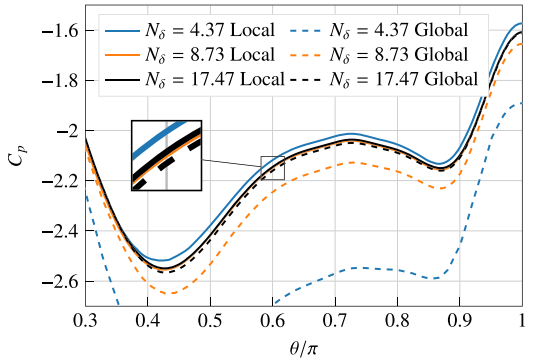
and are in close agreement with results from the immersed interface method of Gillis et al. [35] and the vortex method of Koumoutsakos and Leonard [42].

Also shown are instantaneous profiles of the non-dimensional surface vorticity $\omega^* = \omega D/U$ as a function of the angular coordinate θ on the cylinder surface ($\theta = 0$ corresponds to the leading stagnation point.) These distributions are taken from the vorticity boundary condition prescribed during the transport step of the discretization. The present vorticity profiles follow closely the results of Marichal [34] at Re = 550 and Wu et al. [59] at Re = 3000. In both cases the profiles also agree well with Gillis et al. [35].

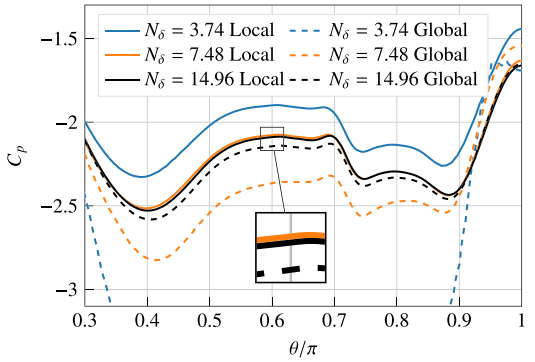
As described in section 5.4, the pressure distribution on the cylinder can be calculated either by integrating the surface vorticity flux (a procedure local to each immersed body) or by solving a pressure Poisson equation (a global elliptic solve). Figs. 14a and 14b show the relative pressure coefficient $C_p(\theta) = 2(p(\theta) - p(0))/u_\infty^2$ resulting from both methods at several spatial resolutions, for Re = 500 and Re = 3000 at $t^* = 2.5$. At comparable values of the N_δ parameter, the convergence of both the local and global pressure calculations at Re = 3000 is slower than at Re = 550. The convergence is more consistent across Reynolds numbers when evaluated with the Q parameter: for both Reynolds numbers, $Q \sim 0.3$ is sufficient for a well-converged local pressure calculation. Similar convergence in the global pressure calculation requires even more resolution, and is only achieved in the finest resolution at Re = 550 ($Q = 0.08$). Figs. 14c and 14d demonstrate that the converged global pressure at Re = 550 and local pressure at Re = 3000 agree well with reference data from the vorticity-based Brinkmann penalization method of Lee et al. [54] and the vorticity-based body-fitted finite volume method of Wu et al. [59].

6.4. Semicircular lid-driven cavity

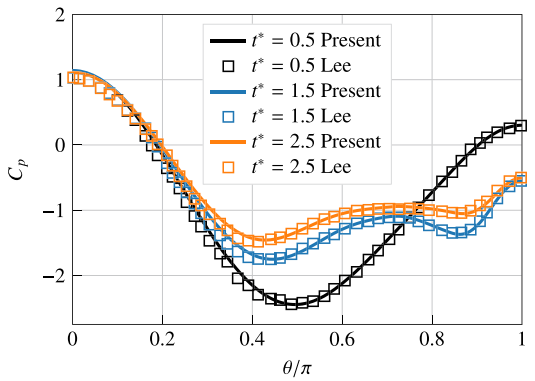
While the method developed here has several computational advantages for exterior flows, it is equally capable of simulating internal flows with concave boundaries. Consider a semicircular cavity with diameter D filled with a stationary fluid of viscosity ν (Fig. 15). At $t = 0$ the top wall of the cavity begins moving rightward with velocity U. The resulting flow is characterized by the Reynolds number $\mathrm{Re} = UD/\nu$, and has been shown to reach a steady state for $\mathrm{Re} \leq 6600$ [60]. To simulate this flow numerically, the semicircular flow domain is embedded in a rectangular computational domain, and the



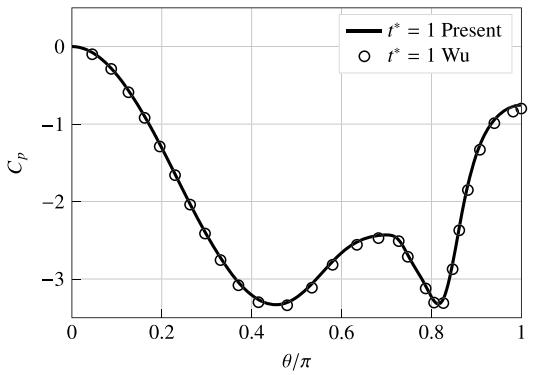
(a) Resolution effects on pressure calculation for Re = 550 and t^* = 2.5. The resolutions considered are N_δ = (4.37, 8.73, 17.47) and Q = (1.23, 0.31, 0.08).



(b) Resolution effects on pressure calculation for Re = 3000 and t^* = 2.5. The resolutions considered are N_δ = (3.74, 7.48, 14.96) and Q = (3.92, 0.98, 0.24).



(c) Surface pressure (global method), Re = 550, N_δ = 17.47.



(d) Surface pressure (local method), Re = 3000, N_δ = 14.96.

Fig. 14. Surface pressure calculations for the impulsively started cylinder at Re = 550 (left column) and $Re = 3000$ (right column).

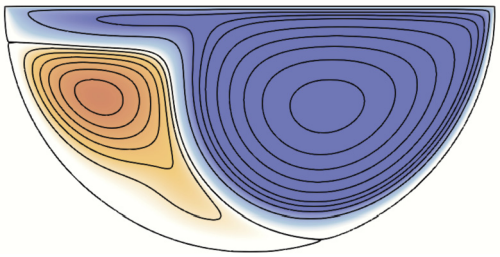


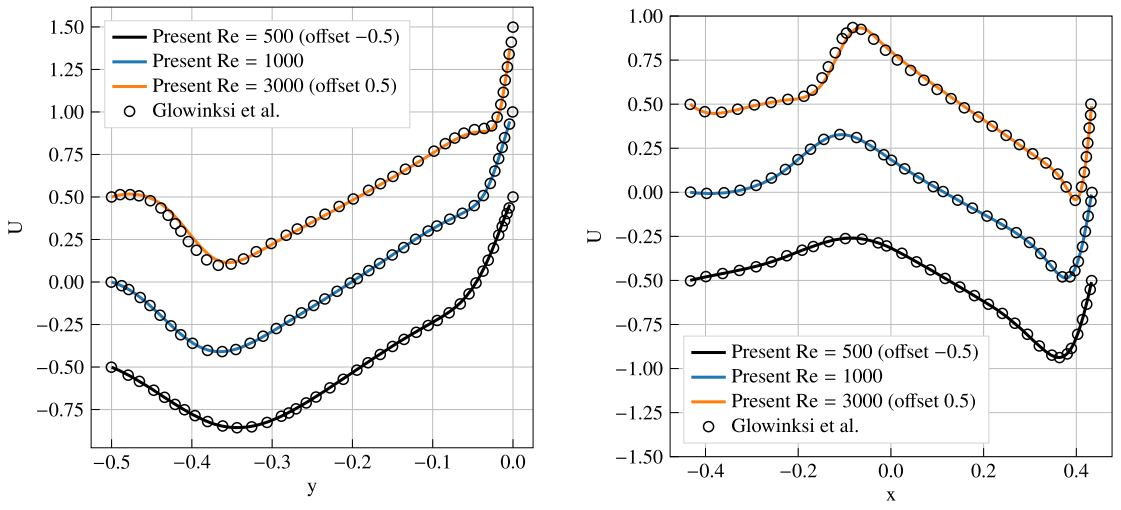
Fig. 15. Steady-state streamlines for the flow in a semicircular lid-driven cavity, colored by stream function ($\mathrm{Re}_D = 3000$). Contours indicate $\psi/UD = (-6\times10^{-3},\ -5\times10^{-3},\ -4\times10^{-3},\ -3\times10^{-3},\ -2\times10^{-3},\ -1\times10^{-3},\ -1\times10^{-6},\ 1\times10^{-6},\ 5\times10^{-3},\ 1\times10^{-2},\ 1.5\times10^{-2},\ 2\times10^{-2},\ 3\times10^{-2},\ 4\times10^{-2},\ 5\times10^{-2},\ 6\times10^{-2},\ 7\times10^{-2})$.

center of the semicircle offset from the grid to break symmetry. The resulting flow is integrated in time until it reaches an approximate steady state. Velocity profiles taken from this steady flow are shown in Fig. 16, and show excellent agreement with those provided by Glowinski et al. [60] for Re = 500, 1000, and 3000.

For internal flows, the boundary of the computational domain lies outside of the fluid domain. As a result, the convolution operator $(\nabla_h^2)^{-1}$ used to solve the velocity reconstruction problem need not satisfy a particular far-field boundary condition. Here we choose the Dirichlet boundary condition $\psi = 0$, which is the most convenient and least computationally expensive option [45].

6.5. Side-by-side cylinder pairs

To validate the ability of our method to simulate flow past multiple bodies, we consider a side-by-side cylinder pair. In this test case, two cylinders of diameter D are placed side-by-side in a free stream flow of velocity U, with centers



(a) Vertical velocities, Re = 500.

(b) Horizontal velocities, Re = 3000.

Fig. 16. Steady-state velocity profiles from the interior of a lid-driven semicircular cavity. Data are offset for readability. Both the (a) vertical and (b) horizontal profiles show excellent agreement with reference data from Glowinski et al. [60] for Reynolds numbers between 500 and 3000.

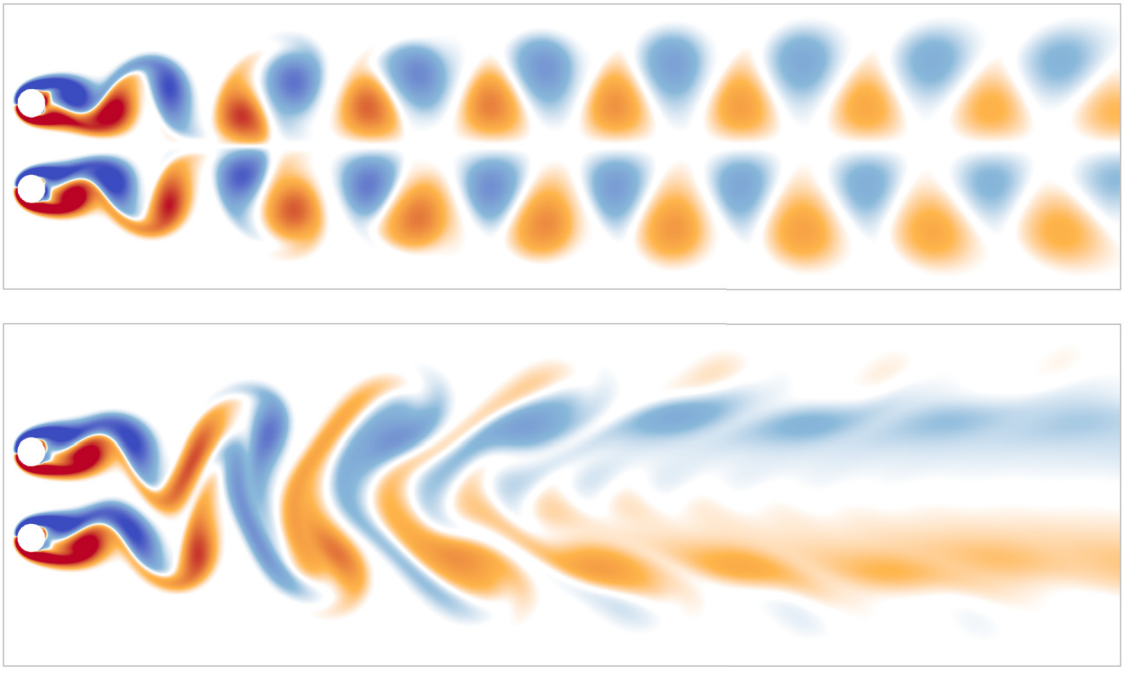


Fig. 17. Two distinct shedding patterns for a side-by-side cylinder pair at $\mathrm{Re} = 100$ and $L/D = 3$: antiphase shedding (top) and in-phase shedding (bottom).

separated by a distance L. The flow is characterized by two non-dimensional parameters, the Reynolds number $\mathrm{Re} = UD/\nu$ and the non-dimensional gap-width L/D. For certain combinations of these two numerical parameters, multiple stable vortex-shedding modes exist [61]; here we consider in-phase and antiphase shedding, illustrated in Fig. 17. Both patterns can be reached from a null initial vorticity field, with antiphase shedding coming from a constant free stream and in-phase shedding coming from a free stream which is initially perturbed to break symmetry.

To simulate this test case, we place the center of each cylinder a distance $1D$ from the inflow boundary. The lack of padding at the inflow is enabled by the true free-space boundary conditions implemented in the velocity reconstruction procedure [45]. At $t = 0$ the flow starts from a null vorticity field, and is integrated in time until a steady-state shedding pattern is reached. The outflow boundary condition described in section 5.2 is prescribed on the downstream domain boundary to allow for long-time integration. An appropriate location for the outflow boundary can be determined by calculating the lift and drag forces resulting from a single set of parameters ($\mathrm{Re} = 100$, $L/D = 3$) and a varied domain length.

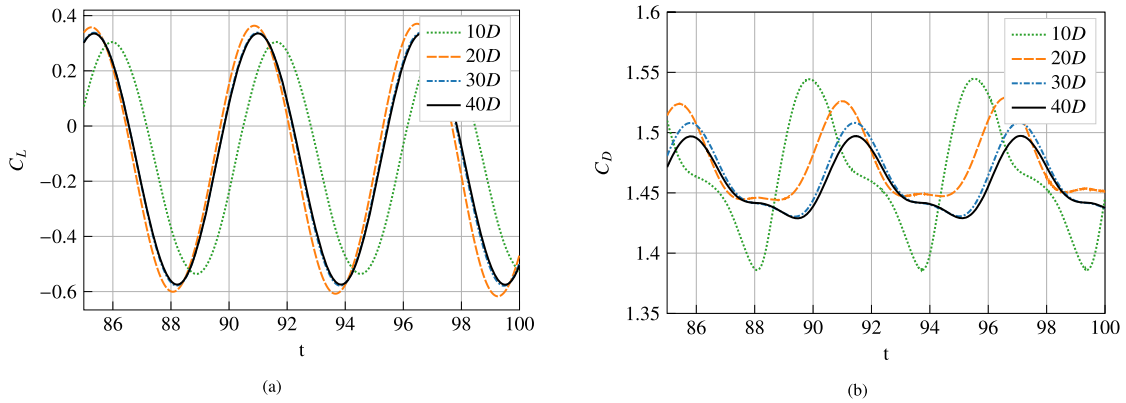


Fig. 18. Time history of the (a) lift coefficient and (b) drag coefficient of the lower cylinder in a cylinder pair at Re = 100 and $L/D = 3$, for different domain lengths. The calculations performed on domains of length $30D$ and $40D$ agree well; shorter domains lead to slight shifts in the amplitude and phase of the lift coefficient, and large qualitative changes in the behavior of the drag coefficient.

Table 1
Long-time statistics of the lift and drag forces on a side-by-side cylinder pair, as calculated Kang [61] and by the present method.

Parameters	Author	$C_{D,Mean}$	$C_{L,Mean}$	$C_{L,RMS}$
Re = 100, $L/D = 3$, Antiphase	Kang	1.46	0.116	0.317
	Present	1.47	0.129	0.319
Re = 100, $L/D = 3$, In-Phase	Kang	1.44	0.129	0.190
	Present	1.42	0.120	0.183
Re = 100, $L/D = 4$, Antiphase	Kang	1.43	0.082	0.280
	Present	1.42	0.070	0.273
Re = 160, $L/D = 3$, Antiphase	Kang	1.45	0.100	0.510
	Present	1.44	0.092	0.507
Re = 160, $L/D = 4$, Antiphase	Kang	1.40	0.058	0.440
	Present	1.39	0.056	0.443

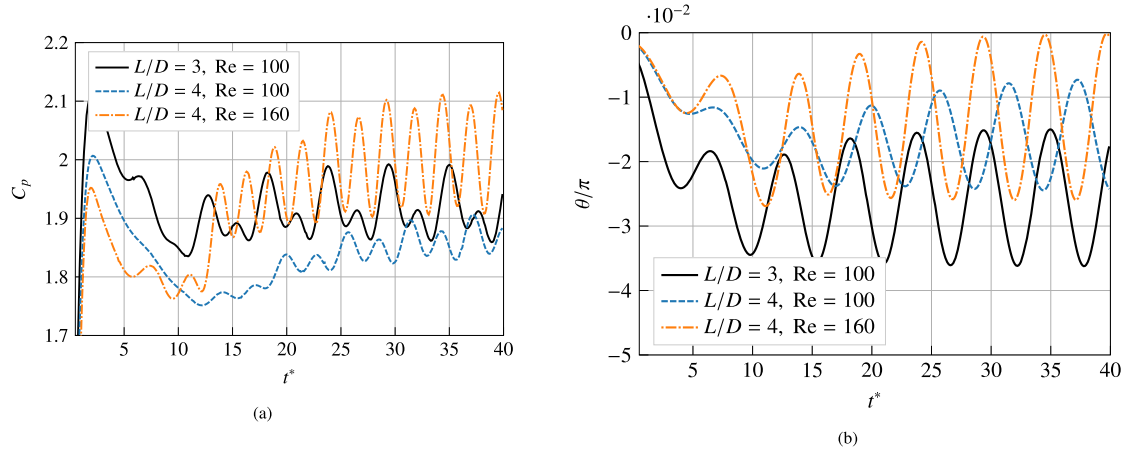


Fig. 19. (a) Time history of the pressure difference between the leading stagnation point and rear $\theta = \pi$ point for cylinder pairs undergoing antiphase shedding at various gap widths and Reynolds numbers. (b) The time history of the location of the leading stagnation point on the lower cylinder for the same set of parameters considered on the left.

The results, shown in Fig. 18, indicate that these forces are relatively insensitive to outflow location for domains longer than $30D$, and to allow a margin of safety we adopt a domain of size $12D \times 40D$. For all of the simulations shown here, the spatial resolution has been chosen to ensure accurate pressure calculations via surface integration. Using the resolution parameters defined in section 6.3, this corresponds to $Q = 0.17$ at Re = 100 and $Q = 0.34$ at Re = 160.

Table 1 lists the steady-state statistics of the drag coefficient $C_D = 2F_x/DU^2$ and lift coefficient $C_L = 2F_y/DU^2$ of each cylinder for a variety of Reynolds numbers, gap widths, and shedding patterns. Reference values are provided by Kang [61],

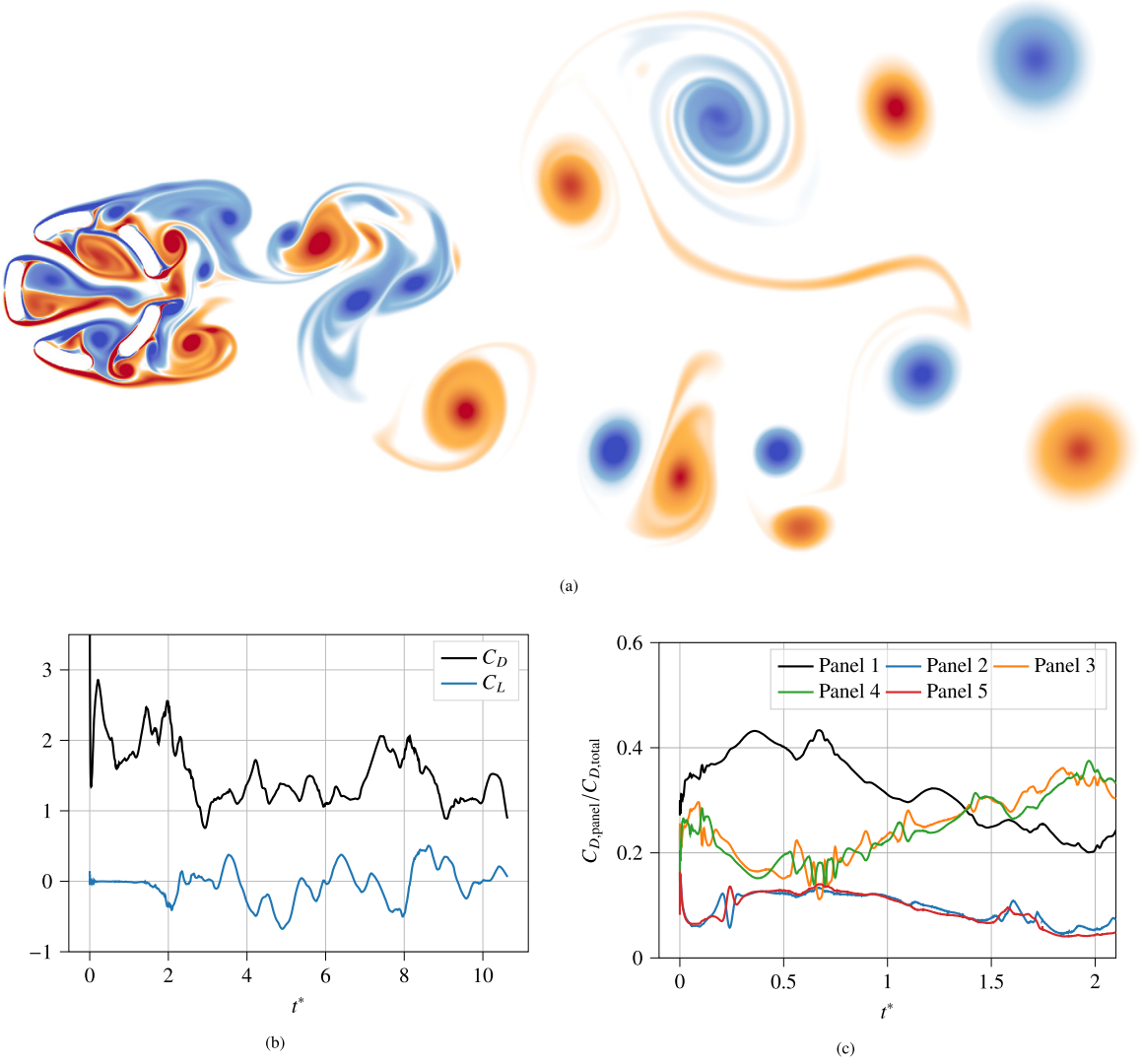


Fig. 20. (a) A snapshot of the vorticity field resulting from flow over a group of non-convex solid bodies inspired by a submerged offshore aquaculture cage structure. The snapshot is recorded at $t^* = U_\infty t/D = 10.0$ after an initial impulsive start. (b) Time history of the total lift and drag coefficients for the aquaculture cage. (c) Early time history of the fraction of the total drag coefficient attributed to each panel of the cage. Panels are numbered counterclockwise beginning with the leftmost panel.

who uses a velocity-pressure immersed boundary method with an $80D \times 100D$ computational domain. The two sets of results show good agreement, despite the fact the present method uses a domain that is sixteen times smaller in area. In addition to lift and drag forces, the use of a sharp immersed method allows for the calculation of time-dependent pressure distributions on the immersed cylinders. Fig. 19a displays the time history of $C_p = 2(p_{stag} - p(\pi))/u_\infty^2$, the difference between the pressure at the leading stagnation point and the $\theta = \pi$ point on the downstream side of each cylinder, for cylinder pairs undergoing antiphase shedding at three different sets of parameters. For these flows the leading stagnation point is identified as the point of zero vorticity closest to $\theta = 0$ on each cylinder, which is calculated from the time-dependent surface vorticity distribution. The time history of the angular location of this point on the lower cylinder of each pair is provided in Fig. 19b.

6.6. Multiple non-convex immersed bodies

We finally demonstrate the flexibility of this framework through a test case that combines multiple non-convex obstacles in an external flow. Fig. 20a provides a snapshot of the vorticity field that results from the flow over a collection of solid bodies inspired by a submerged offshore aquaculture cage structure. Each panel of the cage is a slice of an annulus with a thickness that is one tenth of the full cage diameter D, and each slice covers an angular extent of $\pi/5$ radians excluding the semicircular cap on each end. The simulation is impulsively started with a horizontal free stream velocity of u_∞, and the

Reynolds number based on the outer cage diameter is $\mathrm{Re}_D = u_\infty D/\nu = 4000$. Vorticity is allowed to exit the computational domain via an outflow boundary condition placed a distance of $3.249D$ downstream from the center of the cage. Fig. 20b provides a time history of the lift coefficient $C_L = 2F_y/u_\infty^2 D$ and drag coefficient $C_D = 2F_x/u_\infty^2 D$ over the course of the simulation, where F_x and F_y are the total forces parallel and perpendicular to the free stream. Additionally, Fig. 20c plots the fraction of the total drag coefficient attributed to each of the five panels which make up the cage. The non-dimensional time coordinate in both figures is defined as $t^* = u_\infty t/D$. Throughout the simulation the circulation around each body is automatically tracked by our solver, and our IIM is robust to the geometric issues caused by concave surfaces (as discussed in section 2.2).

7. Conclusion

We have presented a 2D vorticity-based immersed interface method that can simulate fluid flows in bounded and unbounded domains, with multiple non-convex immersed bodies, and outflow boundary conditions. Our approach relies on a re-interpretation of the explicit jump IIM which simplifies the implementation and addresses challenges posed by nonconvex bodies. We show that the use of conservative spatial discretizations allows for the discrete enforcement of Kelvin's theorem, which is essential for simulations with multiple bodies and outflow boundary conditions. Lastly, we have built upon an efficient FFT-accelerated elliptic solver to solve the velocity reconstruction problem with multiple immersed bodies on arbitrary domain topologies. On test cases with known solutions, the resulting method achieves second order spatial convergence in the infinite error norm over the entire domain, and third order temporal convergence. We reproduce reference results for a variety of internal and external flows with Reynolds numbers between 100 and 3000, and accurately recover lift forces, drag forces, moments, and time-dependent traction distributions on immersed solid bodies. Lastly, we have demonstrated that with free-space and outflow boundary conditions, our vorticity-based approach can recover accurate solutions with a domain size that is sixteen times smaller than that of the velocity-based reference results.

We consider several immediate future directions for this work. While this work focuses on simulations with stationary solid bodies, immersed interface methods are well-suited for simulations involving moving and deforming geometry, including fluid-structure interaction problems that are often discretized with lower-order immersed boundary or penalization methods [62,63]. There are also promising developments in extending the IIM to non-smooth geometries with thin features, cusps, and acute interior corners [33], which would further broaden the range of flows that can be simulated with the current method. The accuracy of surface pressure and shear distributions can be greatly increased through the use of multi-resolution adaptive grids, which allow computational elements to be concentrated around immersed surfaces. Finally, such multiresolution adaptive grids would pave the way for a computationally-efficient extension to 3D, building upon [36] as well as the various improvements laid out in this work.

CRediT authorship contribution statement

James Gabbard: Conceptualization, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Thomas Gillis:** Conceptualization, Supervision, Writing – review & editing. **Philippe Chatelain:** Conceptualization, Project administration, Writing – review & editing. **Wim M. van Rees:** Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

James Gabbard reports financial support was provided by Mathworks Inc.

Acknowledgements

We wish to acknowledge financial support from a MathWorks Engineering Fellowship (JG); from a postdoctoral fellowship from the Belgian American Educational Foundation (BAEF) and an International Excellence Scholarship from Wallonie Bruxelles International (TG); from a MIT International Science and Technology Initiatives (MISTI) Seed Fund award (PC, WVR); and from an Early Career Award from the Department of Energy, Program Manager Dr. Steven Lee, award number DE-SC0020998 (JG, TG, WVR).

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jcp.2022.111339.

References

[1] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.

[2] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[3] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2007) 2118–2137.
[4] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numer. Math. 81 (1999) 497–520.
[5] M. Coquerelle, G.-H. Cottet, A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies, J. Comput. Phys. 227 (2008) 9121–9137.
[6] M. Gazzola, P. Chatelain, W.M. van Rees, P. Koumoutsakos, Simulations of single and multiple swimmers with non-divergence free deforming geometries, J. Comput. Phys. 230 (2011) 7093–7114.
[7] M.M. Hejlesen, P. Koumoutsakos, A. Leonard, J.H. Walther, Iterative Brinkman penalization for remeshed vortex methods, J. Comput. Phys. 280 (2015) 547–562.
[8] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. Von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (2008) 4825–4852.
[9] J.H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, J. Comput. Phys. 230 (2011) 7347–7363.
[10] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[11] Z. Li, K. Ito, The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains, SIAM, 2006.
[12] F. Gibou, D. Hyde, R. Fedkiw, Sharp interface approaches and deep learning techniques for multiphase flows, J. Comput. Phys. 380 (2019) 442–463.
[13] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.
[14] D.M. Ingram, D.M. Causon, C.G. Mingham, Developments in Cartesian cut cell methods, Math. Comput. Simul. 61 (2003) 561–572.
[15] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, J. Comput. Phys. 171 (2001) 822–842.
[16] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier-Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.
[17] 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 (2006) 109–138.
[18] A. Wiegmann, K.P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (2000) 827–862.
[19] F. Vilar, C.-W. Shu, Development and stability analysis of the inverse Lax–Wendroff boundary treatment for central compact schemes, ESAIM: Math. Model. Numer. Anal. 49 (2015) 39–67.
[20] T. Li, C.-W. Shu, M. Zhang, Stability analysis of the inverse Lax–Wendroff boundary treatment for high order upwind-biased finite difference schemes, J. Comput. Appl. Math. 299 (2016) 140–158.
[21] T. Li, C.-W. Shu, M. Zhang, Stability analysis of the inverse Lax–Wendroff boundary treatment for high order central difference schemes for diffusion equations, J. Sci. Comput. 70 (2017) 576–607.
[22] T. Li, J. Lu, C.-W. Shu, Stability analysis of inverse Lax–Wendroff boundary treatment of high order compact difference schemes for parabolic equations, J. Comput. Appl. Math. 400 (2022) 113711.
[23] J. Lu, J. Fang, S. Tan, C.-W. Shu, M. Zhang, Inverse Lax–Wendroff procedure for numerical boundary conditions of convection–diffusion equations, J. Comput. Phys. 317 (2016) 276–300.
[24] C. Brehm, H.F. Fasel, A novel concept for the design of immersed interface methods, J. Comput. Phys. 242 (2013) 234–267.
[25] C. Brehm, C. Hader, H.F. Fasel, A locally stabilized immersed boundary method for the compressible Navier-Stokes equations, J. Comput. Phys. 295 (2015) 475–504.
[26] P.T. Brady, D. Livescu, Foundations for high-order, conservative cut-cell methods: stable discretizations on degenerate meshes, J. Comput. Phys. 426 (2021) 109794.
[27] C. Brehm, H. Fasel, Immersed interface method for solving the incompressible Navier-Stokes equations with moving boundaries, in: 49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2011, p. 758.
[28] S. Tan, C.-W. Shu, A high order moving boundary treatment for compressible inviscid flows, J. Comput. Phys. 230 (2011) 6023–6036.
[29] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, J. Comput. Phys. 176 (2002) 231–275.
[30] Z. Li, C. Wang, A fast finite difference method for solving Navier-Stokes equations on irregular domains, Commun. Math. Sci. 1 (2003) 180–196.
[31] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.
[32] S. Hosseinverdi, H.F. Fasel, An efficient, high-order method for solving Poisson equation for immersed boundaries: combination of compact difference and multiscale multigrid methods, J. Comput. Phys. 374 (2018) 912–940.
[33] S. Hosseinverdi, H.F. Fasel, A fourth-order accurate compact difference scheme for solving the three-dimensional Poisson equation with arbitrary boundaries, in: AIAA Scitech 2020 Forum, 2020, p. 0805.
[34] Y. Marichal, An immersed interface vortex particle-mesh method, Ph.D. thesis, UCLouvain, 2014.
[35] T. Gillis, Y. Marichal, G. Winckelmans, P. Chatelain, A 2D immersed interface vortex particle-mesh method, J. Comput. Phys. 394 (2019) 700–718.
[36] T. Gillis, Accurate and efficient treatment of solid boundaries for the vortex particle-mesh method, Ph.D. thesis, UCLouvain, 2019.
[37] M.L. Parks, E. de Sturler, G. Mackey, D.D. Johson, S. Maiti, Recycling Krylov subspaces for sequences of linear systems, SIAM J. Sci. Comput. 28 (2006) 1651–1674.
[38] T. Gillis, G. Winckelmans, P. Chatelain, Fast immersed interface Poisson solver for 3D unbounded problems around arbitrary geometries, J. Comput. Phys. 354 (2018) 403–416.
[39] Y. Marichal, P. Chatelain, G. Winckelmans, Immersed interface interpolation schemes for particle–mesh methods, J. Comput. Phys. 326 (2016) 947–972.
[40] W.H. Press, H. William, S.A. Teukolsky, A. Saul, W.T. Vetterling, B.P. Flannery, Numerical recipes, in: The Art of Scientific Computing, 3rd edition, Cambridge University Press, 2007.
[41] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer, 1998, pp. 325–432.
[42] P. Koumoutsakos, A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, J. Fluid Mech. 296 (1995) 1–38.
[43] L.J. Zhang, J.D. Eldredge, A viscous vortex particle method for deforming bodies with application to biolocomotion, Int. J. Numer. Methods Fluids 59 (2009) 1299–1320.
[44] J. Cantarella, D. DeTurck, H. Gluck, Vector calculus and the topology of domains in 3-space, Am. Math. Mon. 109 (2002) 409–442.
[45] D.-G. Caprace, T. Gillis, P. Chatelain, FLUPS: a Fourier-based library of unbounded Poisson solvers, SIAM J. Sci. Comput. 43 (2021) C31–C60.
[46] Z. Yang, A Cartesian grid method for elliptic boundary value problems in irregular regions, Ph.D. thesis, University of Washington, 1996.
[47] P. Poncet, Analysis of an immersed boundary method for three-dimensional flows in vorticity formulation, J. Comput. Phys. 228 (2009) 7268–7288.
[48] J. Lequeurre, A. Munnier, Vorticity and stream function formulations for the 2D Navier–Stokes equations in a bounded domain, J. Math. Fluid Mech. 22 (2020) 1–73.

[49] P. Chatelain, S. Backaert, G. Winckelmans, S. Kern, Large eddy simulation of wind turbine wakes, Flow Turbul. Combust. 91 (2013) 587–605.
[50] A. Thom, The flow past circular cylinders at low speeds, Proc. R. Soc. Lond. Ser. A, Contain. Pap. Math. Phys. Character 141 (1933) 651–669.
[51] W. E, J.-G. Liu, Vorticity boundary condition and related issues for finite difference schemes, J. Comput. Phys. 124 (1996) 368–382.
[52] L. Quartapelle, Numerical Solution of the Incompressible Navier-Stokes Equations, Birkhäuser, Basel, 1993.
[53] F. Noca, On the evaluation of time-dependent fluid-dynamic forces on bluff bodies, Ph.D. thesis, California Institute of Technology, 1997.
[54] S.J. Lee, J.H. Lee, J.C. Suh, Computation of pressure fields around a two-dimensional circular cylinder using the vortex-in-cell and penalization methods, Model. Simul. Eng. 2014 (2014).
[55] J.H. Williamson, Low-storage Runge-Kutta schemes, J. Comput. Phys. 35 (1980) 48–56.
[56] B. Hejazialhosseini, D. Rossinelli, C. Conti, P. Koumoutsakos, High throughput software for direct numerical simulations of compressible two-phase flows, in: SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, IEEE, 2012, pp. 1–12.
[57] P. Lagerstrom, Laminar Flow Theory, Princeton University Press, Princeton, N.J., 1996.
[58] C.R. Anderson, M.B. Reider, A high order explicit method for the computation of flow about a circular cylinder, J. Comput. Phys. 125 (1996) 207–224.
[59] C. Wu, S.A. Kinnas, Z. Li, Y. Wu, A conservative viscous vorticity method for unsteady unidirectional and oscillatory flow past a circular cylinder, Ocean Eng. 191 (2019).
[60] R. Glowinski, G. Guidoboni, T.-W. Pan, Wall-driven incompressible viscous flow in a two-dimensional semi-circular cavity, J. Comput. Phys. 216 (2006) 76–91.
[61] S. Kang, Characteristics of flow over two circular cylinders in a side-by-side arrangement at low Reynolds numbers, Phys. Fluids 15 (2003) 2486–2498.
[62] C.H. Rycroft, C.-H. Wu, Y. Yu, K. Kamrin, Reference map technique for incompressible fluid–structure interaction, J. Fluid Mech. 898 (2020).
[63] Y. Bhosale, T. Parthasarathy, M. Gazzola, A remeshed vortex method for mixed rigid/soft body fluid–structure interaction, J. Comput. Phys. 444 (2021) 110577.