Improved accuracy of the discrete immersed boundary formulation with regularized delta functions and a well-conditioned form of its projection-based solution

Diederik Beckers*

Graduate Aerospace Laboratories, California Institute of Technology, 1200 East California Boulevard, Pasadena, 91125, CA, USA

H. Jane Bae

Graduate Aerospace Laboratories, California Institute of Technology, 1200 East California Boulevard, Pasadena, 91125, CA, USA

Andres Goza

Grainger College of Engineering, Department of Aerospace Engineering, University of Illinois Urbana-Champaign, 104 S Wright Street, Urbana, 61801, IL, USA

Abstract

The immersed boundary (IB) method with regularized delta functions, pioneered by Peskin [1] is often reported to be only first-order accurate. This low-order accuracy is commonly attributed to smoothing the solution near the interface and interpolating it to the interface using regularized delta functions. We demonstrate that these operations do not prohibit higher accuracy. We leverage a smoothed indicator (or Heaviside) function, as in the multiphase flow method by Tryggvason et al. [2] and the immersed layers method by Eldredge [3], to view the IB solution as a composite of distinct interior and exterior solutions. We demonstrate that this perspective, accounting for additional, higher-order terms that are ignored in previous works, leads to improved accuracy. We empirically show that the accuracy of the proposed method is second order for Poisson problems and slightly less than second order for incompressible Navier-Stokes problems. However, we believe this approach more generally provides a path for exact second-order accuracy (and higher). We show that the proposed methodology can be incorporated into a projection IB formalism. Projection IB methods produce spurious surface stresses through ill-conditioned linear systems that are fragile to the resolution used to represent the immersed surface relative to the flow domain. The proposed method simultaneously addresses this conditioning problem, without heuristic parameters or post-processing, as well as the limitation of these projection IB methods to first-order accuracy.

Keywords: Immersed boundary methods, Higher-order accuracy, Numerical conditioning, Regularized Dirac delta functions, Incompressible Navier-Stokes, Poisson's equation, Projection method

1. Introduction

Immersed boundary (IB) methods are used to solve partial differential equations in a physical domain $\Omega \subset \mathbb{R}^3$ while enforcing interface conditions on an IB: an arbitrarily shaped surface $\Gamma \subset \Omega$ lying strictly inside the boundaries of the computational domain. The key feature of these methods is that the discretization of Ω need not conform to Γ, enabling the use of solvers optimized for Cartesian grids on simple geometries and avoiding expensive mesh (re-)generation, which is particularly advantageous when Γ moves in time.

*Corresponding author

Email address: diederikbeckers@ucla.edu (Diederik Beckers)

In the original IB method of Peskin [1], the effect of Γ is accounted for through a spatially singular term in the governing equation that transforms a surface forcing distribution $f(\boldsymbol{\xi})$, where $\boldsymbol{\xi}$ is a surface parameter, to the solution domain,

$$F(\boldsymbol{x}) = \int_\Gamma f(\boldsymbol{\xi})\delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{\xi}))\mathrm{d}S(\boldsymbol{\xi}), \quad \boldsymbol{x} \in \Omega, \tag{1}$$

where $\delta(\boldsymbol{x})$ is the three-dimensional Dirac delta function (DDF) and $\boldsymbol{X}(\boldsymbol{\xi})$ is the Lagrangian coordinate of a given point on the IB. Similarly, interface conditions at the IB are specified using the sifting property of the Dirac delta function to evaluate the solution at the IB,

$$u(\boldsymbol{\xi}) = \int_\Omega u(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{\xi}))\mathrm{d}\boldsymbol{x}, \quad \boldsymbol{\xi} \in \Gamma. \tag{2}$$

In Peskin's approach, Γ is discretized with Lagrangian markers—material points tracked in time. Since these points generally do not coincide with the Eulerian grid used to discretize Ω, the singular DDF is replaced by a regularized DDF $\delta_{\Delta\boldsymbol{x}}$ with compact support spanning only a few grid cells. This function spreads the Lagrangian forces to nearby Eulerian grid points and is constructed to satisfy $\sum_{i,j,k} \delta_{\Delta\boldsymbol{x}}(\{\mathsf{x}\}_{i,j,k} - \boldsymbol{X}) = 1$ for all $\boldsymbol{X}$, where $\{\mathsf{x}\}_{i,j,k}$ denotes the position of an Eulerian grid point indexed by $i, j, k \in \mathbb{Z}$. In three dimensions with $\boldsymbol{x} = (x, y, z)$ and for a grid with uniform spacings $\Delta\boldsymbol{x} = (\Delta x, \Delta y, \Delta z)$, the regularized DDF is typically the tensor product

$$\delta_{\Delta\boldsymbol{x}}(\boldsymbol{x}) = \frac{1}{\Delta x \Delta y \Delta z}\phi\Big(\frac{x}{\Delta x}\Big)\phi\Big(\frac{y}{\Delta y}\Big)\phi\Big(\frac{z}{\Delta z}\Big), \tag{3}$$

where ϕ is a smooth, compactly supported kernel function.

If we discretize Γ with N_l markers, where $\{\mathbf{X}\}_l$ represents the Lagrangian coordinate of the l-th marker and if $\{\mathrm{S}\}_l$ is its associated surface area, then we can discretize the right-hand side of Eq. (1) as a Riemann sum,

$$\sum_l f(\{\boldsymbol{\xi}\}_l)\delta_{\Delta\boldsymbol{x}}(\boldsymbol{x} - \{\mathbf{X}\}_l)\{\mathrm{S}\}_l, \tag{4}$$

which is commonly referred to as the *regularization* of the IB forcing. Similarly, the right-hand side of Eq. (2) becomes

$$\sum_{i,j,k} u(\boldsymbol{x}_{i,j,k})\delta_{\Delta\boldsymbol{x}}(\boldsymbol{x}_{i,j,k} - X)\Delta x \Delta y \Delta z, \tag{5}$$

corresponding to a discrete convolution-type *interpolation* using $\delta_{\Delta\boldsymbol{x}}$ as the kernel. Throughout the remainder of the paper, we refer to this discretization technique—namely, the approximation of continuous convolution with a DDF kernel by a Riemann sum and regularized DDF—as the *continuous forcing discretization* and we call the methods this approach *continuous forcing IB methods*, following the classification of Mittal and Iaccarino [4].

Both regularization and interpolation introduce errors, and the overall convergence rate depends on the type of IB problem. For regularization, Beyer and LeVeque [5] analyzed the IB method applied to the 1D heat equation with singular forcing. They show that, when the IB force is prescribed, the accuracy of the solution away from the interface is determined by the number of discrete moment conditions satisfied by the regularized DDF. Tornberg and Engquist [6] extended this analysis to multidimensional Poisson problems and demonstrated that the same one-dimensional moment conditions control the accuracy of the multidimensional tensor-product DDF. Specifically, if a regularized, one-dimensional DDF satisfies the q discrete moment conditions

$$\sum_i (\{\mathsf{x}\}_i - X)^r \delta_{\Delta\boldsymbol{x}}(\{\mathsf{x}\}_i - X) = \begin{cases} 1, & r = 0, \\ 0, & 1 \le r \le q, \end{cases} \tag{6}$$

for all X, then, when used with a q-th order discretization of the underlying PDE, the IB method with prescribed forcing converges with order q away from the interface and with first-order accuracy near the interface. This reduction to first order near the interface arises because the singular forcing induces jumps in the solution or its derivatives, which the regularized IB forcing necessarily spreads over several grid points around the interface. As a result, the numerical solution gradually changes from the solution on one side to the other over these grid points, which only matches the sharp exact solution with first order accuracy.

It is important to note this result is not valid when the forcing is unknown a priori and instead has to be determined based on a constraint on the solution that involves interpolating the solution to the interface using Eq. (5). In this setting, the accuracy of the interpolation operation affects the accuracy of the solution. Beyer and LeVeque [5] show that the interpolation with discrete DDFs is dictated by the discrete moment conditions when the interpolation operation is associated with a smooth solution. However, for non-smooth solutions, as are typical when singular IB forcing creates distinct solutions that connect at an interface, the interpolation accuracy is reduced to first order. As the interpolated result is used to constrain the solution at the IB, the global accuracy of the solution is reduced to first order [5]. Note that even a higher order interpolation scheme for non-smooth solutions would still produce a first-order method, because the near-interface solution values used in interpolation would utilize first-order accurate information from the regularized force. The focus of this work is, therefore, to provide a method with greater than first accuracy by addressing these limitations in both the regularization and the interpolation operations.

Some works have addressed the accuracy limitations of the immersed boundary method while retaining the use of regularized DDFs. For example, Griffith and Peskin [7] demonstrated that formally second-order accuracy can be achieved for sufficiently smooth solutions by giving the immersed boundary a finite thickness, which eliminates the jump discontinuities responsible for first-order errors in the classical formulation. Likewise, Stein et al. [8] showed that higher-order accuracy can be obtained by constructing a smooth extension of the solution across the interface, such that the solution varies smoothly over the IB and the accuracy of Eq. (5) is governed by the discrete moment conditions 6. However, both approaches impose structural constraints that reduce the generality of the IB method: the former requires assigning the interface a nonphysical thickness, and the latter effectively suppresses independent physical solution branches on either side of the interface.

The second focus of this work concerns the computation of the IB force. In Peskin's original formulation the immersed boundary is a flexible, infinitesimally-thin membrane whose Lagrangian markers move with the surrounding incompressible viscous fluid. The force applied by the membrane is obtained from a constitutive relation between the marker displacements and stresses in massless elastic fibers connecting the markers. While this approach is natural for flexible structures, it becomes problematic when used for rigid bodies by imposing a large stiffness for the elastic fibers, which can lead to severe stability and stiffness issues in explicit or weakly-coupled time stepping.

Several alternative methods have been proposed that avoid these stability restrictions (see, e.g., the recent reviews of Mittal and Seo [9] and Verzicco [10]). Among these, distributed Lagrange multiplier (DLM) and the IB projection method have proven particularly robust. In these formulations, the IB forces are Lagrange multipliers that enforce the no-penetration and no-slip conditions on the immersed boundary, analogous to the role of pressure as the Lagrange multiplier enforcing incompressibility. However, while these methods avoid the stability restrictions, DLM and projection-based formulations become increasingly ill-conditioned as the ratio $\Delta s/\Delta x$ of the Lagrangian marker spacing Δs to the Eulerian grid spacing Δx, decreases. A tradeoff must be found between numerical conditioning—which deteriorates rapidly for $\Delta s/\Delta x \lesssim 1$—and leakage between Lagrangian points, which becomes significant for $\Delta s/\Delta x > 1$, especially at high Reynolds numbers [11]. Values of $\Delta s/\Delta x$ used in prior studies vary: for example, Taira and Colonius [12] recommend $\Delta s/\Delta x \approx 1$, whereas Kallemov et al. [13] use $\Delta s/\Delta x \approx 2$.

Goza et al. [14] showed that, for these methods that utilize the force as a Lagrange multiplier, ill-conditioning for small $\Delta s/\Delta x$ arises because the solution process involves a linear equation for the surface stress. This equation, when cast using regularized DDFs, is an ill-posed first-kind integral equation; consequently, its discrete counterpart is inherently ill-conditioned. Intuitively, when $\Delta s/\Delta x$ is small, high-frequency variations in the surface-force distribution are smoothed by key ingredients in the IB method: regularization to the grid, solution of the Navier–Stokes equations with this forcing (including diffusion), and interpolation back to the boundary. As a result, the inverse problem is highly sensitive to small errors (e.g., from discretization or roundoff). For example, the correct surface stress to produce a physical flow is hard to separate from a surface stress with the right nominal behavior but spurious deviations about it: the smoothing IB process yields a similar velocity field. The numerical result of this sensitivity is that Lagrange multiplier-based IB methods produce stresses with incorrect high-frequency oscillations, computed from ill-conditioned linear systems that are more expensive to solve via iterative techniques than well-conditioned problems. Goza et al. [14] proposed a postprocessing technique to smooth out this noise; however, it does not reduce the intrinsic cost of solving the system. Other approaches to mitigating the ill-posedness have also been applied: Kallemov et al. [13] employ a preconditioner but still face constraints on the $\Delta s/\Delta x$ ratio, while Yu and Pantano [15] use Tikhonov regularization, which requires selecting an ad-hoc regularization parameter.

In this work, we simultaneously address the accuracy challenge associated with continuous forcing IB methods

and the ill-conditioning of the projection-based approach for satisfying Dirichlet boundary conditions (such as the no-slip condition). The proposed method is framed within a Lagrange multiplier-based formulation for the surface stress, to enable a robust treatment of flows past bodies moving with arbitrary prescribed kinematics. It retains a comparable computational cost per timestep to the first-order variants of that method, removes the ill conditioning of the linear system for the surface stress, and does not rely on any ad-hoc parameters. The approach formally distinguishes the solution on either side of the immersed interface, using indicator (Heaviside) functions, a common approach in multiphase methods [16, 17], and we follow the approach of Tryggvason et al. [2] and, in particular, Eldredge [3] to connect it to the immersed boundary method. A Taylor series is then utilized to establish the regularization and interpolation conditions across the interface, which simultaneously improves the accuracy of the operations and produces a second-kind (well posed) linear system for the unknown Lagrange multiplier. We present this approach first for a simple Dirichlet Poisson problem in Section 2 and afterward for the incompressible Navier-Stokes equations with no-slip boundary conditions in Section 3. Although we demonstrate the improved accuracy only using the projection-based approach to find the forces, our methods are applicable to any form of the IB with regularized DDFs.

2. A higher-order accurate, well-conditioned projection IB method for a model Dirichlet Poisson problem

The surface Γ partitions the domain Ω into two subregions, Ω^+ and Ω^-, which we refer to as the *exterior* and *interior* regions, respectively. We will make use of the unit normal vector $\boldsymbol{n}$ and the two tangent vectors $\boldsymbol{t}$ and $\boldsymbol{b}$ defined on Γ. By convention, the normal vector $\boldsymbol{n}$ is oriented to point into Ω^+. An example is shown in the left image of Fig. 1. We also define the indicator functions $H^+(\boldsymbol{x})$ and $H^-(\boldsymbol{x})$ associated with the exterior and interior subdomains, respectively, where $H^\pm(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in \Omega^\pm$ and $H^\pm(\boldsymbol{x}) = 0$ otherwise. In the distributional sense, the gradient of the indicator fields is a distribution of multidimensional Dirac delta functions supported on the interface multiplied by the interface normal,

$$\nabla H^\pm(\boldsymbol{x}) = \pm \int_\Gamma \boldsymbol{n}(\boldsymbol{\xi})\delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{\xi}))\mathrm{d}S(\boldsymbol{\xi}). \tag{7}$$

We develop our method first for a Dirichlet Poisson problem on the global domain Ω for a scalar field $u(x) \in \mathbb{R}$ that is defined as the composite (or piecewise) field,

$$u \coloneqq H^+u^+ + H^-u^-, \tag{8}$$

where u^+ and u^- are the exterior and interior solutions, respectively, each satisfying the Dirichlet Poisson problems

$$\nabla^2 u^\pm(\boldsymbol{x}) = q(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega^\pm, \tag{9}$$

$$u^\pm(\boldsymbol{x}) = u^\pm_{\partial\Omega}, \quad \boldsymbol{x} \in \partial\Omega^\pm \setminus \Gamma, \tag{10}$$

$$u^\pm(\boldsymbol{X}(\boldsymbol{\xi})) = u^\pm_\Gamma(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \Gamma, \tag{11}$$

where q is an integrable source function, $u^\pm_{\partial\Omega}$ is a prescribed boundary condition at the boundary of the exterior and interior domains (excluding Γ), and $u^\pm_\Gamma$ is a prescribed function on Γ. For concreteness, we are only treating the case where the interface condition on the IB is a Dirichlet condition that is the same for the exterior and interior solutions, i.e., $u^\pm_\Gamma = u_\Gamma$, resulting in a continuous solution. However, the method we develop can easily be adapted to account for discontinuities in the solution or different types of interface conditions, such as Neumann conditions.

By applying the product rule for the gradient and divergence operators consecutively to the composite solution (8), we obtain a governing equation for u:

$$\nabla^2 u = q + \nabla H^+ \cdot (\nabla u^+ - \nabla u^-) + \nabla \cdot \Big(\nabla H^+ (u^+ - u^-)\Big), \quad \boldsymbol{x} \in \Omega. \tag{12}$$

Using (7), the properties of the DDF, and the fact that $u^+_\Gamma = u^-_\Gamma$, one can further simplify (12) to a Poisson's equation with a single-layer source term [3] and formulate the Dirichlet Poisson problem (8)-(11) as:

$$\nabla^2 u = q + \int_\Gamma \sigma(\boldsymbol{\xi})\delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{\xi}))\mathrm{d}S(\boldsymbol{\xi}), \quad \boldsymbol{x} \in \Omega, \tag{13}$$

$$u(\boldsymbol{x}) = u_{\partial\Omega}(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega, \tag{14}$$

$$\int_\Omega u(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{\xi}))\mathrm{d}\boldsymbol{x} = u_\Gamma(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \Gamma, \tag{15}$$

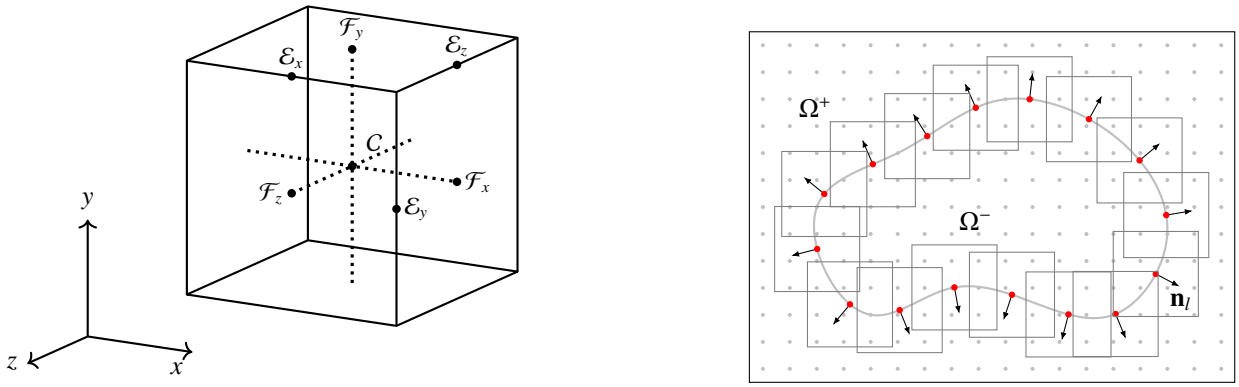

Figure 1: (Left) Three-dimensional grid cell. (Right) Two-dimensional example of a discretized surface immersed in a rectangular domain.

where $u_{\partial\Omega}$ is the prescribed boundary condition at the boundary of the entire solution domain. $\sigma(\boldsymbol{\xi}) = (u_\Gamma^{n+}(\boldsymbol{\xi}) - u_\Gamma^{n-}(\boldsymbol{\xi}))$ is the single layer strength density, with $u_\Gamma^{n\pm}(\boldsymbol{\xi}) := \boldsymbol{n}(\boldsymbol{\xi}) \cdot \nabla u(\boldsymbol{X}(\boldsymbol{\xi}))$ the normal derivative of the exterior and interior solutions at the interface.

Unlike most continuous forcing IB methods, we do not directly discretize the continuous governing IB equations such as (13). Rather, we follow an approach similar to that of Eldredge [3], where we start from the discrete analogue of (8) and apply discrete product rules to obtain the discrete analogue of Eq. (12). This ensures that our formulation is consistent with the computation of the discrete indicator fields that we will make use of later. However, while Eldredge's formulation then reduces back to a continuous forcing discretization of Eq. (13) when $u_\Gamma^+ = u_\Gamma^-$, we modify his derivation to retain additional terms that increase the accuracy of the method.

2.1. Discrete problem formulation for a composite solution

We discretize Ω using a three-dimensional, staggered Cartesian grid that does not necessarily conform to Γ. The grid has constant cell dimensions $(\Delta x, \Delta y, \Delta z)$ and contains three sets of discrete locations (see the right panel of Fig. 1): cell centers C, cell faces $\mathcal{F} = (\mathcal{F}_x, \mathcal{F}_y, \mathcal{F}_z)$, and cell edges $\mathcal{E} = (\mathcal{E}_x, \mathcal{E}_y, \mathcal{E}_z)$. We define the vector spaces $\mathbb{R}^C$, $\mathbb{R}^{\mathcal{F}_x}$, $\mathbb{R}^{\mathcal{F}_y}$, $\mathbb{R}^{\mathcal{F}_z}$, $\mathbb{R}^{\mathcal{E}_x}$, $\mathbb{R}^{\mathcal{E}_y}$, and $\mathbb{R}^{\mathcal{E}_z}$ for the fields discretized at those cell locations, and will use the notation $\mathbb{R}^{\mathcal{F}} := \mathbb{R}^{\mathcal{F}_x \times \mathcal{F}_y \times \mathcal{F}_z}$ and $\mathbb{R}^{\mathcal{E}} := \mathbb{R}^{\mathcal{E}_x \times \mathcal{E}_y \times \mathcal{E}_z}$ for discrete spaces containing all three components of vector-valued fields. For example, the data structure containing all the elements of a discretized field of scalar data (such as pressure) on the cell centers is in $\mathbb{R}^C$. Similarly, the data structure containing all elements of all three components of a discretized vector field (such as velocity) on the cell faces is in $\mathbb{R}^{\mathcal{F}}$. The cell edges store the results of curl operations applied to vector fields defined on cell faces, and in the following section they are also used to store tensor-valued data. Additionally, we describe the positions of the cell centers by $\mathbf{x}_C = (\mathsf{x}_C, \mathsf{y}_C, \mathsf{z}_C) \in \mathbb{R}^{C \times C \times C}$, with $\mathsf{x}_C, \mathsf{y}_C, \mathsf{z}_C \in \mathbb{R}^C$, and similarly for the faces and edges, for example, $\mathbf{x}_{\mathcal{F}_x} = (\mathsf{x}_{\mathcal{F}_x}, \mathsf{y}_{\mathcal{F}_x}, \mathsf{z}_{\mathcal{F}_x}) \in \mathbb{R}^{\mathcal{F}_x \times \mathcal{F}_x \times \mathcal{F}_x}$, with $\mathsf{x}_{\mathcal{F}_x}, \mathsf{y}_{\mathcal{F}_x}, \mathsf{z}_{\mathcal{F}_x} \in \mathbb{R}^{\mathcal{F}_x}$. Throughout the rest of this paper we will use second-order finite differences to discretize our differential operators.

We define the discrete indicator fields $\mathsf{H}_C^\pm \in \mathbb{R}^C$ and the discrete composite solution $\overline{\mathsf{u}} \in \mathbb{R}^C$ on the cell centers,

$$\overline{\mathsf{u}} = \mathsf{H}_C^+ \circ \mathsf{u}^+ + \mathsf{H}_C^- \circ \mathsf{u}^-, \tag{16}$$

where $\mathsf{u}^\pm \in \mathbb{R}^C$ are the discrete interior and exterior solutions and $\circ$ is the element-wise product between two fields defined on the same space. We then aim to formulate a system of equations for this composite solution such that the discrete interior and exterior solutions satisfy Eq. (9)-(10) when discretized on our grid,

$$\mathsf{L}\mathsf{u}^\pm = \mathsf{q}^\pm + \mathsf{b}, \tag{17}$$

and also satisfy the interface conditions at the points that discretize Γ. Here, $\mathsf{L} : \mathbb{R}^C \mapsto \mathbb{R}^C$ is the discrete Laplacian and b represents the boundary-condition term that arises from discretizing the Laplacian while enforcing Eq. (10). We use the overbar notation in Eq. (16) to indicate the composite nature of the solution and the potential smoothing near the IB, which depends on whether the discrete indicator functions are themselves smoothed. This contrasts with the analytic solution (8), which is expressed using sharp indicator functions.

When the indicator functions are smoothed, the solution from each side will bleed into the other side and we need an extension of the solution over that support. One of the key aspects of our method is that we enforce that $u^{\pm}$ vary smoothly across the IB. Away from the IB, and outside of their respective regions, their extension is irrelevant provided it is finite, well-defined, and will still make the solution satisfy the governing equation and interface condition. The proposed approach works directly with the composite solution that stitches together the exterior and interior solutions, so that the superfluous interior (exterior) part of the exterior (interior) solution is not needed, in contrast to the method of Stein et al. [8].

We can multiply Eq. (17) for the interior and exterior solutions by their respective indicator fields and add the two equations:

$$\mathsf{H}_C^+ \circ \mathsf{L}\mathsf{u}^+ + \mathsf{H}_C^- \circ \mathsf{L}\mathsf{u}^- = \overline{\mathsf{q}} + \mathsf{b} \tag{18}$$

where $\overline{\mathsf{q}} \coloneqq \mathsf{H}_C^+ \circ \mathsf{q}^+ + \mathsf{H}_C^- \circ \mathsf{q}^-$ and we required that $\mathsf{H}_C^+ + \mathsf{H}_C^- = 1$. Then, using the discrete product rule (A.29) for the Laplacian, we can recast this into a governing equation for $\overline{\mathsf{u}}$ that is the discrete equivalent of Eq. (12):

$$\mathsf{L}\overline{\mathsf{u}} = \overline{\mathsf{q}} + \mathsf{b} + {}^{C}\mathsf{I}_{\mathcal{F}}\left(\mathsf{G}\mathsf{H}_C^+ \circ (\mathsf{G}\mathsf{u}^+ - \mathsf{G}\mathsf{u}^-)\right) + \mathsf{D}\left(\mathsf{G}\mathsf{H}_C^+ \circ {}^{\mathcal{F}}\mathsf{I}_C\,(\mathsf{u}^+ - \mathsf{u}^-)\right), \tag{19}$$

where $\mathsf{D} : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{C}$ and $\mathsf{G} : \mathbb{R}^{C} \mapsto \mathbb{R}^{\mathcal{F}}$ are the discrete divergence and gradient operators, respectively. The operation ${}^{\mathcal{F}}\mathsf{I}_C : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{C}$ transforms a vector on the cell faces to a scalar on the cell centers by interpolating each vector component to the cell center and adding them together, and ${}^{C}\mathsf{I}_{\mathcal{F}} : \mathbb{R}^{C} \mapsto \mathbb{R}^{\mathcal{F}}$ is the adjoint operation, which interpolates a scalar on the cell faces to each vector component on the cell faces (see Appendix A.1 for more details). This equation is not yet in a convenient form for solving because it involves both the composite and the individual solutions. However, the gradient of the indicator function is zero everywhere except at the interface, where it becomes unbounded for sharp indicator functions. For smoothed indicator functions—where the smoothing is limited to a finite-width region around the interface—the gradient remains nonzero but finite in that region. We can take advantage of this compact support to rewrite the terms in Eq. (19) that involve the interior and exterior solutions so that they depend only on their behavior at the interface. Moreover, because of this compact support, discretizing the divergence and gradient operators does not introduce any additional boundary-condition terms.

2.2. Finding the discrete indicator fields and their gradients from the regularized DDF

At this stage—and only at this stage—we apply the continuous forcing discretization formula (4) to find the discrete analogue of the relation (7) between the gradient of the indicator fields and the DDF. To proceed, we first introduce a more detailed notation tailored to our grid.

We discretize Γ into a collection of N_l surface patches whose centers we track as Lagrangian points. We discretize both scalar- and vector-valued surface quantities on these points and define the spaces for these data structures as $\mathbb{R}^{\mathcal{S}}$ and $\mathbb{R}^{\mathcal{V}}$, respectively. The positions of the points are described by $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}) \in \mathbb{R}^{\mathcal{V}}$ and the local normal and tangential vectors by $\mathbf{n}, \mathbf{t}, \mathbf{b} \in \mathbb{R}^{\mathcal{V}}$. The areas of the surface patches are given by $\mathrm{S} \in \mathbb{R}^{\mathcal{S}}$. We use the notation $\circ$ to denote element-wise multiplication between these quantities. This operation applies component-by-component multiplication between arrays of matching type: for scalar-valued arrays, it corresponds to the standard Hadamard product; for vector-valued arrays, it multiplies corresponding vector components. We also apply it between scalar and vector arrays, in which case it acts as a point-wise scalar–vector multiplication.

To simplify our notation, we define the field $\mathsf{d}_{C,l} \in \mathbb{R}^{C}$ that results from evaluating $\delta_{\Delta \boldsymbol{x}}$ centered at the l-th surface point at the grid points:

$$\mathsf{d}_{C,l} \coloneqq \delta_{\Delta \boldsymbol{x}}\Big(\mathsf{x}_C - \{\mathrm{X}\}_l, \mathsf{y}_C - \{\mathrm{Y}\}_l, \mathsf{z}_C - \{\mathrm{Z}\}_l\Big), \tag{20}$$

and we define $\mathsf{d}_{\mathcal{F}_x,l}$, $\mathsf{d}_{\mathcal{F}_y,l}$, and $\mathsf{d}_{\mathcal{F}_z,l}$ similarly. Using these fields, we can define the regularization operators $\mathsf{R}_C : \mathbb{R}^{\mathcal{S}} \mapsto \mathbb{R}^{C}$ and $\mathsf{R}_{\mathcal{F}} : \mathbb{R}^{\mathcal{V}} \mapsto \mathbb{R}^{\mathcal{F}}$ to regularize a surface scalar $\mathrm{s} \in \mathbb{R}^{\mathcal{S}}$ to the cell centers and a surface vector $\mathbf{v} = (\mathrm{v}_x, \mathrm{v}_y, \mathrm{v}_z) \in \mathbb{R}^{\mathcal{V}}$ to the cell faces by summing over all the discrete surface points:

$$\{\mathsf{R}_C \mathrm{s}\}_{i,j,k} \coloneqq \sum_l \{\mathsf{d}_{C,l}\}_{i,j,k}\{\mathrm{s}\}_l\{\mathrm{S}\}_l, \qquad \{\mathsf{R}_{\mathcal{F}}\mathbf{v}\}_{i,j,k} \coloneqq \sum_l \begin{pmatrix} \{\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{v}_x\}_l \\ \{\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{v}_y\}_l \\ \{\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{v}_z\}_l \end{pmatrix} \{\mathrm{S}\}_l, \tag{21}$$

which are the discrete scalar and vector versions of Eq. (1), specialized for our staggered grid. Similarly, we can discretize Eq. (2) and (15) for our grid using the operators $\mathsf{E}_C : \mathbb{R}^{\mathcal{C}} \mapsto \mathbb{R}^{\mathcal{S}}$ and $\mathsf{E}_{\mathcal{F}} : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{\mathcal{V}}$ to interpolate a scalar field $\mathsf{s} \in \mathbb{R}^{\mathcal{C}}$ and vector field $\mathbf{v} = (\mathsf{v}_x, \mathsf{v}_y, \mathsf{v}_z) \in \mathbb{R}^{\mathcal{F}}$ to the IB points, where, for the l-th body point,

$$\{\mathsf{E}_C \mathsf{s}\}_l \coloneqq \Delta x \Delta y \Delta z \sum_{i,j,k} \{\mathsf{d}_{C,l}\}_{i,j,k} \{\mathsf{s}\}_{i,j,k}, \qquad \{\mathsf{E}_{\mathcal{F}} \mathbf{v}\}_l \coloneqq \Delta x \Delta y \Delta z \sum_{i,j,k} \begin{pmatrix} \{\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k} \{\mathsf{v}_x\}_{i,j,k} \\ \{\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k} \{\mathsf{v}_y\}_{i,j,k} \\ \{\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k} \{\mathsf{v}_z\}_{i,j,k} \end{pmatrix}. \tag{22}$$

Using the previous notation, we can discretize the relation (7) between the indicator fields and DDF as [18, 17, 3]:

$$\mathsf{G}\mathsf{H}_C^{\pm} = \pm \mathsf{R}_{\mathcal{F}} \mathbf{n}. \tag{23}$$

If we apply the discrete divergence operation to both sides and require that our discrete operators satisfy mimetic properties [19], we obtain the following discrete Poisson equation [18, 17, 3] involving the discrete Laplacian (since $\mathsf{D}\mathsf{G} = \mathsf{L}$):

$$\mathsf{L}\mathsf{H}_C^{\pm} = \pm \mathsf{D}\mathsf{R}_{\mathcal{F}} \mathbf{n} + \mathsf{b}_{\mathsf{H}}. \tag{24}$$

By applying appropriate Dirichlet boundary conditions b_{H}, we can solve for the indicator field whose smoothness is determined by the smoothness of the regularized DDF used in (20) and which satisfies $\mathsf{H}_C^+ + \mathsf{H}_C^- = 1$. However, it is important to note that if the formula for the multidimensional regularized DDF is not curl-free—as is the case for most functions proposed for IB methods—then the discrete indicator field obtained by solving Eq. (24) will, in general, not satisfy Eq. (23) for curved surfaces. Instead, the gradient satisfies

$$\mathsf{G}\mathsf{H}_C^{\pm} = \pm \mathsf{R}_{\mathcal{F}} \mathbf{n} \pm \mathsf{C}\mathsf{L}_{\mathcal{E}}^{-1} \mathsf{C}^{\top} \mathsf{R}_{\mathcal{F}} \mathbf{n}, \tag{25}$$

where $\mathsf{L}_{\mathcal{E}} : \mathbb{R}^{\mathcal{E}} \mapsto \mathbb{R}^{\mathcal{E}}$ is the discrete Laplacian for variables on edges and $\mathsf{C} : \mathbb{R}^{\mathcal{E}} \mapsto \mathbb{R}^{\mathcal{F}}$ and $\mathsf{C}^{\top} : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{\mathcal{E}}$ are the discrete curl operation and its adjoint, respectively. We will ignore this last term in the rest of our derivation, which will be a source of error in our global solution. To avoid this, one can use a curl-free DDF, which in our experience would require a custom treatment for each body point. Alternatively, one can attempt to find an improved H^+ that satisfies Eq. (23) through a modified Poisson's equation that includes information about the curvature of the surface. This topic, including quantifying the error from ignoring the last term in (25), is part of our ongoing investigation.

If we substitute Eq. (23) into Eq. (19), we obtain

$$\mathsf{L}\overline{\mathsf{u}} = \overline{\mathsf{q}} + \mathsf{b} + {}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\Big(\mathsf{R}_{\mathcal{F}} \mathbf{n} \circ (\mathsf{G}\mathsf{u}^+ - \mathsf{G}\mathsf{u}^-)\Big) + \mathsf{D}\Big(\mathsf{R}_{\mathcal{F}} \mathbf{n} \circ {}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}(\mathsf{u}^+ - \mathsf{u}^-)\Big), \tag{26}$$

which still contains the interior and exterior solutions, but now along with the DDF, which is only nonzero is a small region at the interface that scales with the grid spacing. The last step to formulate our governing equation for $\overline{\mathsf{u}}$ is to replace the interior and exterior solutions (and their gradients) by their behavior inside this region using a Taylor series expansion.

2.3. Using Taylor series to find higher-order terms

When the DDFs are centered at the discrete surface points and element-wise multiplied by a field variable, we can approximate the variation of this variable within the support of the discrete DDF by a truncated Taylor series expansion about the discrete surface point at which the DDF is centered. For example, for our solution variable, we can use (A.36) to formulate

$$\begin{aligned} &\{\mathsf{d}_{C,l} \circ \mathsf{u}^{\pm}\}_{i,j,k} \\ &\quad \approx \{\mathsf{d}_{C,l}\}_{i,j,k} \Big(\{\mathsf{u}_{\Gamma}^{\pm}\}_l + \{\mathbf{n}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l) \{\mathsf{u}_{\Gamma}^{n\pm}\}_l + \{\mathbf{t}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l) \{\mathsf{u}_{\Gamma}^{t\pm}\}_l + \{\mathbf{b}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l) \{\mathsf{u}_{\Gamma}^{b\pm}\}_l \Big), \end{aligned} \tag{27}$$

where $\mathsf{u}, \mathsf{u}^n, \mathsf{u}^t, \mathsf{u}^b \in \mathbb{R}^{\mathcal{S}}$ denote the discrete approximations to u and its normal and tangential derivatives on IB points, respectively.

We use the notation $[\mathrm{s}]_\Gamma \coloneqq \mathrm{s}_\Gamma^+ - \mathrm{s}_\Gamma^-$ to represent the difference between the exterior and interior values (or the *jump*) of a variable $\mathrm{s} \in \mathbb{R}^{\mathcal{S}}$ at the IB. We note that, since $\mathrm{u}_\Gamma^+ = \mathrm{u}_\Gamma^-$ in our case, $[\mathrm{u}]_\Gamma = 0$ and also $[\mathrm{u}^t]_\Gamma = 0$ and $[\mathrm{u}^b]_\Gamma = 0$. Then, using (A.37), we can approximate the expression in parentheses in the third term of the right-hand side of Eq. (26) as

$$\begin{aligned}\{\mathsf{R}_{\mathcal{F}}\mathbf{n} \circ (\mathsf{G}\mathrm{u}^+ - \mathsf{G}\mathrm{u}^-)\}_{i,j,k} &\approx \sum_l \begin{pmatrix} \{\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{n}_x\}_l\{[\mathrm{u}^n]_\Gamma\}_l\{\mathrm{n}_x\}_l \\ \{\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{n}_y\}_l\{[\mathrm{u}^n]_\Gamma\}_l\{\mathrm{n}_y\}_l \\ \{\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{n}_z\}_l\{[\mathrm{u}^n]_\Gamma\}_l\{\mathrm{n}_z\}_l \end{pmatrix}\{\mathrm{S}\}_l \\ &= \mathsf{R}_{\mathcal{F}}(\mathbf{n} \circ \mathbf{n} \circ [\mathrm{u}^n]_\Gamma). \end{aligned} \tag{28}$$

Similarly, and because ${}^{\mathcal{F}}\mathsf{I}_C\mathbf{x}_C = \mathsf{x}_{\mathcal{F}}$, we can approximate the part in parentheses in the fourth term in Eq. (26) as

$$\begin{aligned}\{\mathsf{R}_{\mathcal{F}}\mathbf{n} \circ {}^{\mathcal{F}}\mathsf{I}_C(\mathrm{u}^+ - \mathrm{u}^-)\}_{i,j,k} &\approx \sum_l \begin{pmatrix} \{\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{n}_x\}_l\Big(\{\mathbf{n}\}_l \cdot (\mathbf{x}_{\mathcal{F}_x} - \{\mathbf{X}\}_l)\{[\mathrm{u}^n]_\Gamma\}_l\Big) \\ \{\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{n}_y\}_l\Big(\{\mathbf{n}\}_l \cdot (\mathbf{x}_{\mathcal{F}_y} - \{\mathbf{X}\}_l)\{[\mathrm{u}^n]_\Gamma\}_l\Big) \\ \{\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{n}_z\}_l\Big(\{\mathbf{n}\}_l \cdot (\mathbf{x}_{\mathcal{F}_z} - \{\mathbf{X}\}_l)\{[\mathrm{u}^n]_\Gamma\}_l\Big) \end{pmatrix}\{\mathrm{S}\}_l \\ &= \{\mathsf{R}_{\mathcal{F},1n}(\mathbf{n} \circ [\mathrm{u}^n]_\Gamma)\}_{i,j,k}, \end{aligned} \tag{29}$$

where we introduced the shorthand $\mathsf{R}_{\mathcal{F},1n} : \mathbb{R}^{\mathcal{S}} \mapsto \mathbb{R}^{\mathcal{C}}$ for the normal-distance-weighted regularization operation defined in (A.40).

With the approximations (28) and (29), our final formulation for Eq. (26) becomes

$$\mathsf{L}\overline{\mathrm{u}} = \overline{\mathrm{q}} + \mathrm{b} + {}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\mathsf{R}_{\mathcal{F}}(\mathbf{n} \circ \mathbf{n} \circ [\mathrm{u}^n]_\Gamma) + \mathsf{D}\Big(\mathsf{R}_{\mathcal{F},1n}(\mathbf{n} \circ [\mathrm{u}^n]_\Gamma)\Big). \tag{30}$$

We can contrast this with the formulation one would obtain by applying the standard continuous forcing discretization technique on the continuous IB formulation (13):

$$\mathsf{L}\overline{\mathrm{u}} = \overline{\mathrm{q}} + \mathrm{b} + \mathsf{R}_C\mathrm{f}. \tag{31}$$

Because it is known that, in this case, the forcing represents the jump in the normal derivative of the solution [20, 3], we will also refer to $\mathrm{f} = [\mathrm{u}^n]_\Gamma$ as the (IB) forcing strength.

Equation (31) is the same formulation that results from the derivation by Eldredge [3] when $[u]_\Gamma = 0$. Eldredge recognizes that the third and fourth term on the right-hand side of Eq. (26) are the discrete equivalents of the single- and double-layer terms, respectively, from potential theory. Unlike our approach, he replaces ${}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\Big(\mathsf{R}_{\mathcal{F}}\mathbf{n} \circ (\mathsf{G}\mathrm{u}^+ - \mathsf{G}\mathrm{u}^-)\Big)$ by $\mathsf{R}_C[\mathrm{u}^n]_\Gamma$ and $\mathsf{R}_{\mathcal{F}}\mathbf{n} \circ {}^{\mathcal{F}}\mathsf{I}_C(\mathrm{u}^+ - \mathrm{u}^-)$ by $\mathsf{R}_{\mathcal{F}}(\mathbf{n} \circ [\mathrm{u}]_\Gamma)$. While the latter term is zero and ignores the effect of our fourth term on the right-hand side of Eq. (30), the former term is also notably different from our third term. Our third term ${}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\mathsf{R}_{\mathcal{F}}(\mathbf{n} \circ \mathbf{n} \circ [\mathrm{u}^n]_\Gamma)$ might resemble a scaled dot product of the normal vector with itself for each body point, with the result (i.e., $[\mathrm{u}^n]_\Gamma$) placed on the cell centers, which would be equivalent to $\mathsf{R}_C[\mathrm{u}^n]_\Gamma$. However, it is important to note that the DDF evaluated at the cell centers is not exactly equal to the DDF evaluated at the cell faces and, subsequently, interpolated to the cell centers, and we found empirically that the use of R_C limits the global accuracy to first order.

Lastly, we can also use a truncated Taylor series expansion about the IB points for the interior and exterior solutions to analyze the error of the interpolation of the composite solution to the IB points using formula (22):

$$\begin{aligned}&\{\mathrm{u}_\Gamma - \mathsf{E}_C\overline{\mathrm{u}}\}_l \\ &\quad= (\mathrm{u}_\Gamma)_l - \Delta x\Delta y\Delta z \sum_{i,j,k}\{\mathsf{d}_{C,l}\}_{i,j,k}\Big(\{\mathsf{H}_C^+ \circ \mathrm{u}^+\}_{i,j,k} + \{\mathsf{H}_C^- \circ \mathrm{u}^-\}_{i,j,k}\Big) \\ &\quad= (\mathrm{u}_\Gamma)_l - \Delta x\Delta y\Delta z \sum_{i,j,k}\{\mathsf{d}_{C,l}\}_{i,j,k}\Big[\{\mathsf{H}_C^+\}_{i,j,k}\Big(\{\mathrm{u}_\Gamma^+\}_l + \{\mathbf{n}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\{\mathrm{u}_\Gamma^{n+}\}_l\Big) \\ &\qquad+ \{\mathsf{H}_C^-\}_{i,j,k}\Big(\{\mathrm{u}_\Gamma^-\}_l + \{\mathbf{n}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\{\mathrm{u}_\Gamma^{n-}\}_l\Big) + O(\Delta x^2) + O(\Delta y^2) + O(\Delta z^2)\Big]. \end{aligned} \tag{32}$$

Using the zeroth-order discrete moment condition for the DDF (i.e., Eq. (6) with $q = 0$) and the fact that $\mathsf{H}_C^+ + \mathsf{H}_C^- = 1$, u_Γ will cancel out with the first term of the interior and exterior solutions. Additionally, for DDFs that satisfy the

first-order discrete moment condition, we have that

$$\sum_{i,j,k}(\mathsf{d}_{C,l})_{i,j,k}\{\mathbf{n}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l) = 0, \tag{33}$$

for all IB points. Then, using $\mathsf{H}_C^- = 1 - \mathsf{H}_C^+$, we can formulate the following second-order accurate interpolation for our composite solution,

$$\mathsf{E}_C\overline{\mathsf{u}} - [\mathsf{u}^n]_\Gamma \circ \mathsf{E}_{C,1n}\mathsf{H}_C^+ \approx \mathsf{u}_\Gamma, \tag{34}$$

where we introduced the normal-distance-weighted interpolation operation $\mathsf{E}_{C,1n} : \mathbb{R}^C \mapsto \mathbb{R}^S$, defined as:

$$\{\mathsf{E}_{C,1n}\mathsf{s}\}_l \coloneqq \Delta x\Delta y\Delta z \sum_{i,j,k}\{\mathsf{d}_{C,l}\}_{i,j,k}\mathbf{n}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\{\mathsf{s}\}_{i,j,k}. \tag{35}$$

2.4. Projection-based approach to the force computation

The discrete Poisson's equation with the original IB treatment (31) and the uncorrected constraint equation (22) form a constrained system for the composite solution (16), which is a discrete equivalent for the Dirichlet problem (13)-(15). We can express this system in a saddle-point form [21] as:

$$\begin{bmatrix}\mathsf{L} & \mathsf{R}_C\\ \mathsf{E}_C & 0\end{bmatrix}\begin{pmatrix}\overline{\mathsf{u}}\\ -[\mathsf{u}^n]_\Gamma\end{pmatrix} = \begin{pmatrix}\overline{\mathsf{q}} + \mathsf{b}\\ \mathsf{u}_\Gamma\end{pmatrix}. \tag{36}$$

We can find a solution for this system by applying the Schur complement reduction method (see Appendix B) to obtain the following algorithm:

$$\mathsf{L}\overline{\mathsf{u}}^* = \overline{\mathsf{q}} + \mathsf{b}, \tag{37}$$

$$S\,[\mathsf{u}^n]_\Gamma = -(\mathsf{u}_\Gamma - \mathsf{E}_C\overline{\mathsf{u}}^*), \tag{38}$$

$$\overline{\mathsf{u}} = \overline{\mathsf{u}}^* + \mathsf{L}^{-1}\mathsf{R}_C[\mathsf{u}^n]_\Gamma, \tag{39}$$

where the Schur complement of the system (36) is

$$S = \mathsf{E}_C\mathsf{L}^{-1}\mathsf{R}_C. \tag{40}$$

We can also formulate a saddle-point system for our proposed IB method consisting of (30) and the corrected interpolation formula (34), by replacing the element-wise products denoted by $\circ$ with multiplications by diagonal matrices,

$$\begin{bmatrix}\mathsf{L} & {}^C\mathsf{I}_\mathcal{F}\mathsf{R}_\mathcal{F}\mathrm{diag}(\mathbf{n})^2 + \mathsf{D}\mathsf{R}_{\mathcal{F},1n}\mathrm{diag}(\mathbf{n})\\ \mathsf{E}_C & \mathrm{diag}(\mathsf{E}_{C,1n}\mathsf{H}_C^+)\end{bmatrix}\begin{pmatrix}\overline{\mathsf{u}}\\ -[\mathsf{u}^n]_\Gamma\end{pmatrix} = \begin{pmatrix}\overline{\mathsf{q}} + \mathsf{b}\\ \mathsf{u}_\Gamma\end{pmatrix}. \tag{41}$$

The solution for this system is then:

$$\mathsf{L}\overline{\mathsf{u}}^* = (\overline{\mathsf{q}} + \mathsf{b}), \tag{42}$$

$$S\,[\mathsf{u}^n]_\Gamma = -(\mathsf{u}_\Gamma - \mathsf{E}_C\overline{\mathsf{u}}^*), \tag{43}$$

$$\overline{\mathsf{u}} = \overline{\mathsf{u}}^* + \mathsf{L}^{-1}({}^C\mathsf{I}_\mathcal{F}\mathsf{R}_{\mathcal{F},n}\mathrm{diag}(\mathbf{n})^2 + \mathsf{D}\mathsf{R}_{\mathcal{F},1n}\mathrm{diag}(\mathbf{n}))[\mathsf{u}^n]_\Gamma, \tag{44}$$

where the Schur complement of the system (41) is

$$S = -\mathrm{diag}(\mathsf{E}_{C,1n}\mathsf{H}_C^+) - \mathsf{E}_C\mathsf{L}^{-1}({}^C\mathsf{I}_\mathcal{F}\mathsf{R}_\mathcal{F}\mathrm{diag}(\mathbf{n})^2 + \mathsf{D}\mathsf{R}_{\mathcal{F},1n}\mathrm{diag}(\mathbf{n})). \tag{45}$$

This two-step solution strategy is commonly used in immersed methods. For example, Taira and Colonius [12] use it to solve the IB formulation of the Navier-Stokes equations for rigid bodies in velocity-pressure formulation and streamwise-vorticity formulation [22], and they referred to their method as the *immersed boundary projection method* (IBPM), after the projection method for solving the incompressible Navier-Stokes equations. The solution (37)-(39) of the original system is the IBPM formulated for the Poisson's equation [14, 23]. A similar strategy was also used earlier

in several versions of the immersed interface method (IIM) for Poisson-type problems [24, 20] and Navier–Stokes problems in a streamfunction–vorticity formulation [25]. Similar to their method, the unknown jumps in our method (specifically the first derivative in the current Dirichlet Poisson problem) appear in the constraint equation, which fills in the lower right-hand block of the system (41).

Even though both the original and the proposed system use the Schur complement reduction method, their conditioning is very different. Goza et al. [14] show that the original forcing equation with S given by (40) represents the discrete analogue of a Fredholm integral equation of the first kind, which is ill-posed. On the other hand, the new forcing equation with S given by (45) represents the discrete analogue of a well-posed Fredholm integral equation of the second kind, which Li [24] and Wiegmann [20] recognized was also the case for their IIM formulation. The key difference between the original and proposed IB projection formulation in terms of the conditioning lies in the extra diagonal term, which inserts the unknown jump directly in the equation without the IB operators acting on it.

In summary, our modifications to the IB method simultaneously improve the accuracy of the method and the conditioning when used in a projection-based formulation. The overall algorithmic complexity of the proposed method is the same as the original IBPM, and, as a result of the improved conditioning, the method is in many cases faster than the original IBPM, especially when an iterative solver is used to solve Eq. (38) and Eq. (43). We will now apply both the new and original IB method for this Dirichlet Poisson problem to a 1D and 2D example to demonstrate the differences in accuracy and conditioning.

2.5. Results for a 1D example

To better understand the role of the higher-order terms in the accuracy of the IB formulation, we first focus on a 1D example where the Poisson's equation turns into a second-order ODE,

$$\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} = q(x) \quad x \in [x_L, x_R], \tag{46}$$

$$u(x_L) = u_L, \quad u(x_R) = u_R, \quad u(x_\Gamma) = u_\Gamma, \tag{47}$$

where we use $x_L = 0$, $x_R = 2$, $q(x) = -4$, $u^L = u^R = 0$, and the interface condition $u_\Gamma = 0$ at $x_\Gamma = 1$, which we treat with a single IB point with its normal pointing along the x-axis in the positive direction. We will thus consider the subdomain $[x_L, x_\Gamma]$ to be the interior and $[x_\Gamma, x_R]$ to be the exterior. The analytical solution for this problem has a jump in its first derivative $[\mathrm{d}u/\mathrm{d}x]_\Gamma = 4$ at $x = x_\Gamma$. We construct our operators with second-order accurate finite difference schemes and use the smoothed three-point delta function from Yang et al. [26], which satisfies two discrete moment conditions and is non-negative like most other DDFs used in IB methods.

In Fig. 2 we compare three different solutions and their forcing: the solution of the original IB formulation (31) using the analytical first derivative jump as a prescribed IB forcing, the solution of the original IB system (36) where solve for the first-derivative jump such that the solution satisfies the discrete constraint, and the solution of the proposed IB system (36) where we also solve for the first-derivative jump. In Fig. 3 we compare the error of each solution with respect to the analytical solution $u(\mathrm{x})$ evaluated at the same grid points, with and without including the error near the interface, as well as the error of the first-derivative jump at the IB (i.e., the IB forcing).

As expected from the analysis of Tornberg and Engquist [6], the first solution $\overline{\mathrm{u}}_{\mathrm{AF}}$, which uses the analytical first-derivative jump as its forcing, is second-order accurate away from the interface and only first-order accurate within the support of the DDF. However, applying the discrete interpolation (22) does not recover the prescribed interface value u_Γ. Because smoothed DDFs cause the IB method to overestimate the solution near a positive first-derivative jump (and underestimate it for a negative one), the DDF-weighted interpolation yields values that in this case are biased high, leading to an overestimation of the interpolated interface value. Moreover, in cases of interest to us the IB forcing will not generally be known—it must be computed to enforce the constraint—so this approach used for illustrative purposes is inaccessible.

The second solution, $\overline{\mathrm{u}}_{\mathrm{IB1}}$, uses the first-derivative jump computed from (38), which is slightly larger than the analytical one. This follows from the need to satisfy the discrete constraint: for a positive jump, the method must underestimate the solution within the DDF support, and for a negative jump it must overestimate it. As a result, the solution near the interface consists either of values equal to the imposed interface value (possible with a hat-function DDF) or of mixed values that balance to yield the correct DDF-weighted average. In the present example, the positive jump forces an underestimate of the solution, producing a larger effective first-derivative jump. This inability of the

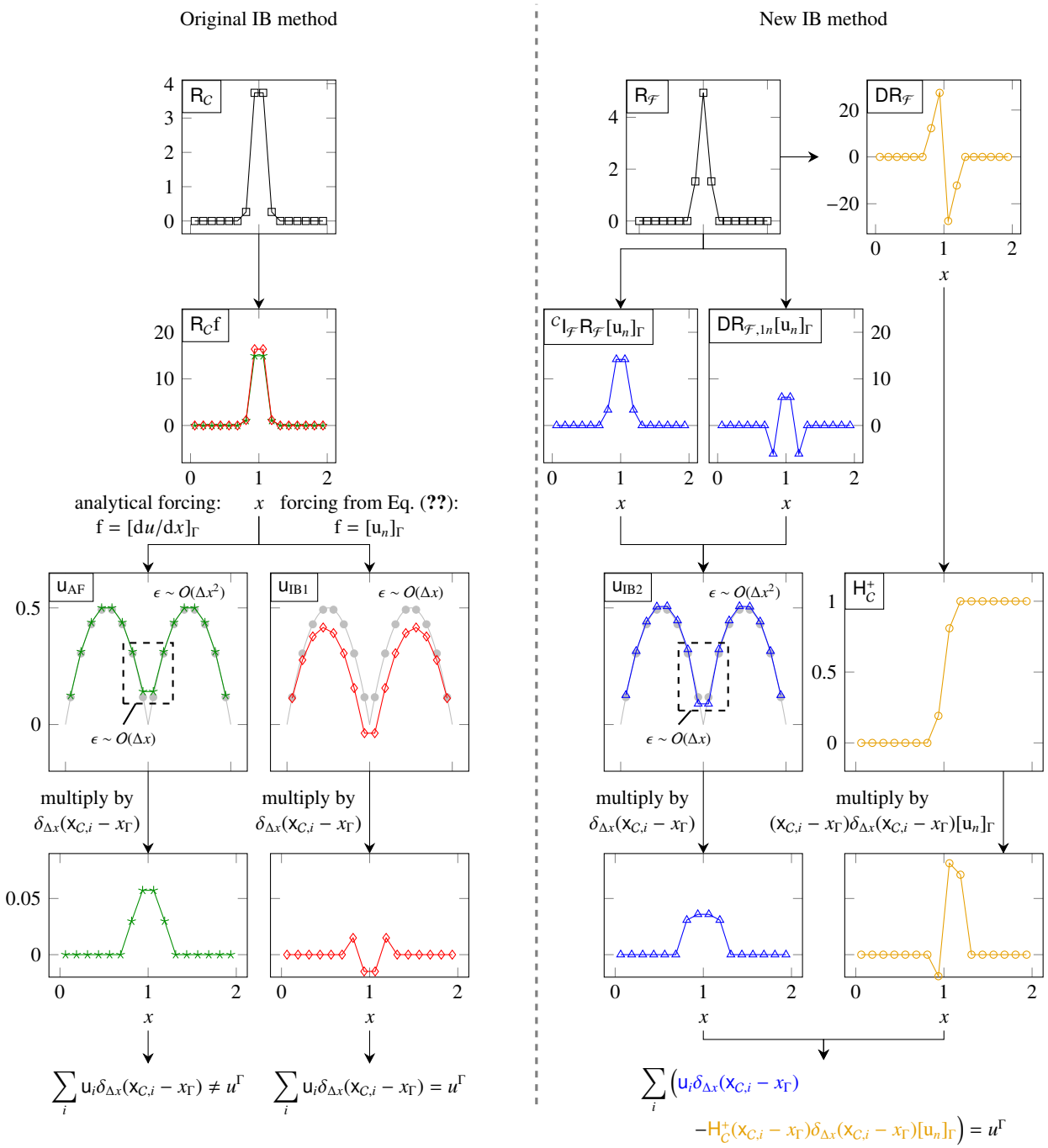

Figure 2: Comparison of three different IB solutions on 16 grid cells for the second-order ODE $\mathrm{d}^2u/\mathrm{d}x^2 = q$ on $[0, 2]$ with $u(0) = u(2) = 0$, $q = -4$, and an IB point at $x_\Gamma = 1$ enforcing $u^\Gamma = 0$. The solutions u_{AF} and $\mathsf{u}_{\mathrm{IB1}}$ on the left of the dashed line use the original IB equation (31) with the analytical forcing strength (green, stars) and the forcing strength obtained through Eq. (38) (red, diamonds), respectively. The solution $\mathsf{u}_{\mathrm{IB1}}$ (blue, triangles) on the right side of the dashed line uses the proposed IB equation (30) with the forcing strength obtained through Eq. (43), such that the discrete constraint (34) is satisfied. The exact solution is shown in gray with filled circles.

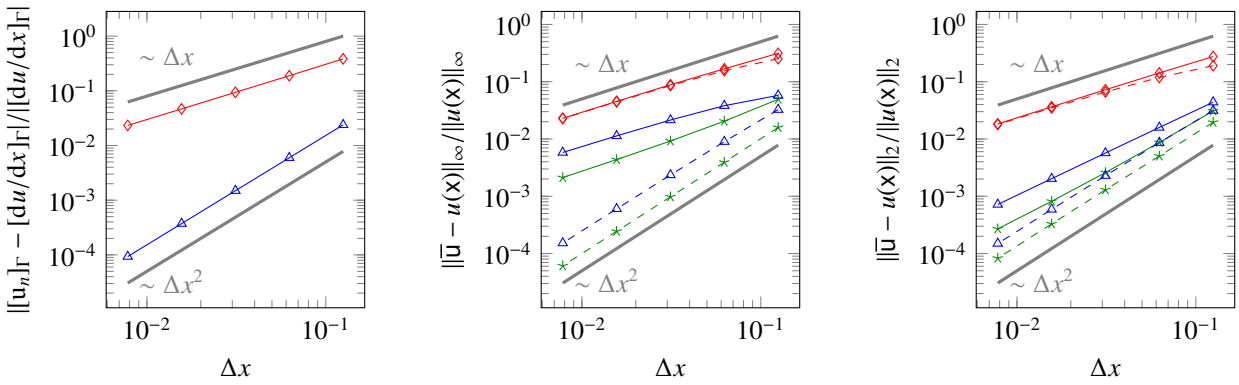

Figure 3: (left) Error of the forcing strength for the 1D Dirichlet Poisson example with respect to to its analytical value for the original IB method (red, diamonds) and the proposed method (blue, triangles). Infinity norm (center) and 2-norm (right) of the numerical solution error computed over the entire domain (solid lines) and over the domain excluding cells within two grid spacings from the interface (dashed lines) for the 1D Dirichlet Poisson example using the original IB equation (31) with the analytical forcing (green, stars) and the forcing strength obtained through Eq. (38) (red, diamonds), and the proposed IB equation (30) with the forcing strength obtained through Eq. (43) (blue, triangles). First- and second-order error scales are also shown (gray, solid).

interpolation to represent a derivative jump is what leads to the characteristic over- and underestimation and ultimately limits both the interpolation and the overall method to first-order accuracy [5].

Lastly, the third solution $\bar{\mathsf{u}}_{\mathrm{IB2}}$, obtained with our proposed IB method, is still first-order accurate within the support of the (interpolated) DDF, but is second-order accurate away from the IB while simultaneously satisfying the discrete constraint equation (34), which involves the indicator field obtained from the same DDF that is used in the governing equation. The second-order accuracy away from the IB can also be obtained by applying the analytical forcing using only the ${}^{C}\mathsf{I}_{\mathcal{F}}\mathsf{R}_{\mathcal{F}}\mathsf{f}$ term. The term $\mathsf{DR}_{\mathcal{F},1n}\mathsf{f}$ has no effect on the solution away from the IB. Its only purpose is to adapt the solution near the IB such that the Taylor series used to obtain (34) are valid.

2.6. Results for a 2D example

Now we apply the previous IB methods to the 2D example of a circle with radius R and outward pointing normals centered at the origin in an unbounded domain, where the boundary condition (10) is replaced by $u(\boldsymbol{x}) \to 0$ as $\boldsymbol{x} \to \infty$. In our discrete method, we can treat this problem using the lattice Green's function on a bounded grid [19]. The prescribed interface condition on Γ is $u_\Gamma(\boldsymbol{\xi}) = \boldsymbol{X}(\boldsymbol{\xi})$, in which case the analytical interior solution (inside the circle) is $u^-(\boldsymbol{x}) = x$ and the exterior solution (outside the circle) in cylindrical coordinates is $u(r,\theta) = R^2/r\cos(\theta)$, and Fig. 4 shows the composite analytical solution. The analytical normal-derivative jump is $[u_n(\boldsymbol{\xi})]_\Gamma = -2\cos(\theta(\boldsymbol{\xi}))$.

We discretize the problem on the finite domain $\Omega = \left\{(x,y) \in \mathbb{R}^2 \;\middle|\; -2R \le x \le 2R,\ -2R \le y \le 2R\right\}$ using several grid sizes and with a body-to-grid spacing ratio $\Delta s/\Delta x \approx 1$, and we compare the resulting solutions of the original and proposed IB method on the left panel of Fig. 4. Figure 5 shows the relative error of the forcing $[\mathsf{u}^n]_\Gamma$ over all the IB points and the error of the solution $\bar{\mathsf{u}}$ over all the grid points or all the grid points excluding the points within three grid spacings of the interface with respect to their analytical counterparts evaluated at the same IB points and grid points. Even though the forcing solution does not converge past a certain grid size, the solutions themselves converge with first-order accuracy when using the infinity norm of the error, and the proposed method's solution converges with second-order accuracy when not considering the point near the interface.

The convergence of the forcing error is also affected by the conditioning of the method, which is shown in the left panel of Fig. 6. As discussed in Sec. 2.4, the forcing equations in the original and proposed method are discrete versions of ill- and well-posed problems, respectively. Although a numerical solution can still be computed using the original method for large values of the body-to-grid spacing $\Delta s/\Delta x$, the problem becomes more ill-conditioned as this value decreases and the discrete surface starts to represent a continuous surface. At around $\Delta s/\Delta x \approx 1$, the conditioning worsens rapidly, and high-frequency numerical noise appears in the forcing solution, as is visible on the right panel of Fig. 6.

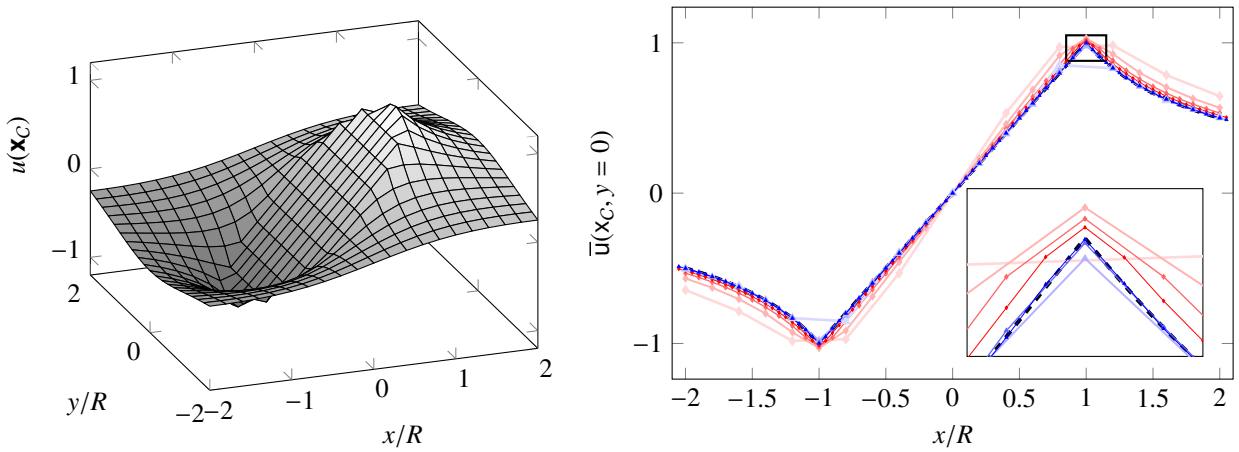

Figure 4: (Left) Analytical solution to the 2D Poisson equation on an unbounded domain subject to a sinusoidal Dirichlet interface condition on a circle centered at the origin with radius R, evaluated at the cell centers of a grid with $\Delta x/R = \Delta y/R = 0.2$. (Right) Comparison of the numerical solution from the proposed method (blue, dashed) with solution from the original IB method (red, dotted) and the exact solution (light gray, solid) on the line along $y = 0$ near the interface for $\Delta s/\Delta x \approx 1$ and four different grid cell sizes $\Delta x/R$: 0.4 (darkest, thick), 0.2, 0.1, and 0.05 (lightest, thin).

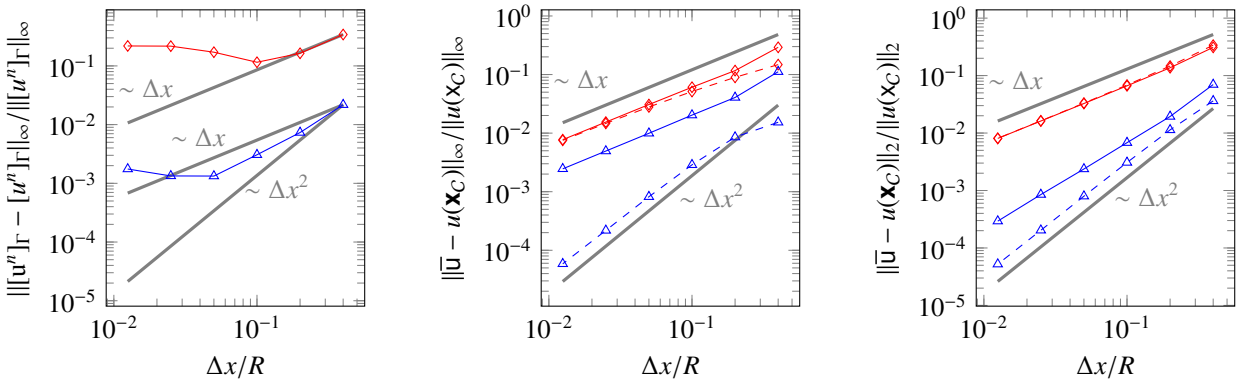

Figure 5: (Left) Infinity norm of the forcing strength error for the 2D Dirichlet Poisson example with respect to to its analytical value for the original IB method (red, diamonds) and the proposed method (blue, triangles). Infinity norm (center) and 2-norm (right) of the numerical solution error computed over all cell centers (solid lines) and over all cell centers excluding those within three grid spacings from the interface (dashed lines) for the 2D Dirichlet Poisson example using the original IB method (red, diamonds) and the proposed method (blue, triangles). First- and second-order error scales are also shown (gray, solid).

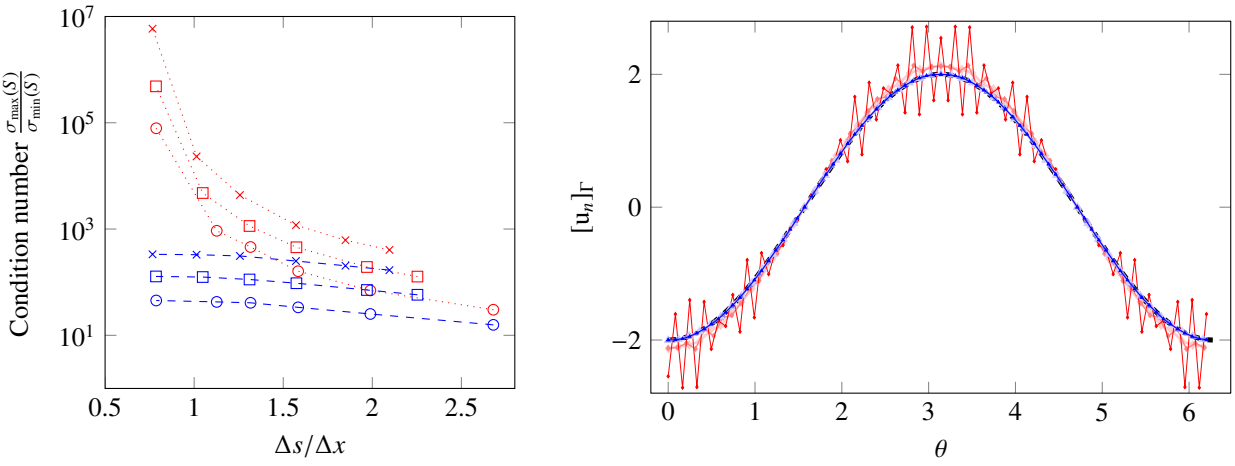

Figure 6: (Left) Condition number of S versus the surface-to-grid spacing ratio for the proposed method (blue, dashed) and the original IB method (red, dotted) using three different grid cell sizes $\Delta x/R$: 0.2 (circles), 0.1 (squares), and 0.05 (crosses). (Right) Comparison of the forcing solution from the proposed method (blue, dashed) with forcing solution from the original IB method (red, dotted) and the exact forcing solution (black, dashed) for $\Delta x/R = 0.2$ and three approximate values of $\Delta s/\Delta x$: 1.21 (lightest, thick), 1.05, and 0.83 (darkest, thin).

3. Treatment of the incompressible Navier-Stokes equations

In this section we derive our new formulation for the non-dimensionalized, incompressible Navier-Stokes equations for the velocity $\boldsymbol{v}^\pm(\boldsymbol{x},t) \in \mathbb{R}^3$ and pressure $p^\pm(\boldsymbol{x},t) \in \mathbb{R}$ in the interior and exterior regions with uniform properties,

$$\frac{\partial \boldsymbol{v}^\pm(\boldsymbol{x},t)}{\partial t} + \nabla\cdot(\boldsymbol{v}^\pm(\boldsymbol{x},t)\boldsymbol{v}^\pm(\boldsymbol{x},t)) = -\nabla p^\pm(\boldsymbol{x},t) + \frac{1}{\mathrm{Re}}\nabla^2\boldsymbol{v}^\pm(\boldsymbol{x},t), \quad \boldsymbol{x}\in\Omega^\pm \tag{48}$$

$$\nabla\cdot\boldsymbol{v}^\pm(\boldsymbol{x},t) = 0, \quad \boldsymbol{x}\in\Omega^\pm \tag{49}$$

$$\boldsymbol{v}^\pm(\boldsymbol{x},t) = \boldsymbol{v}^\pm_{\partial\Omega}(\boldsymbol{x},t), \quad \boldsymbol{x}\in\partial\Omega^\pm\setminus\Gamma(t), \tag{50}$$

$$\boldsymbol{v}^\pm(\boldsymbol{X}(\boldsymbol{\xi},t),t) = \boldsymbol{v}_\Gamma(\boldsymbol{\xi},t), \quad \boldsymbol{\xi}\in\Gamma(t), \tag{51}$$

where Re is the Reynolds number, $\boldsymbol{v}^\pm_{\partial\Omega}$ is a prescribed boundary condition, and $\boldsymbol{v}_\Gamma(\boldsymbol{\xi},t)$ is the prescribed velocity on the IB at the Lagrangian coordinate $\boldsymbol{X}(\boldsymbol{\xi})$ and time t.

Following again the approach of Eldredge [3], we define the composite velocity $\boldsymbol{v} \coloneqq H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-$ and pressure $p = H^+p^+ + H^-p^-$ and apply the relevant product rules and indicator function identities to obtain (see Appendix Appendix C):

$$\frac{\partial \boldsymbol{v}}{\partial t} + \nabla\cdot(\boldsymbol{v}\boldsymbol{v}) - \boldsymbol{F}_{\mathrm{d}\boldsymbol{v}/\mathrm{d}t} - \boldsymbol{F}_{\nabla\cdot(\boldsymbol{v}\boldsymbol{v})} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2\boldsymbol{v} + \boldsymbol{F}_{\nabla p} - \frac{1}{\mathrm{Re}}\boldsymbol{F}_{\nabla^2\boldsymbol{v}}, \quad \boldsymbol{x}\in\Omega, \tag{52}$$

$$\nabla\cdot\boldsymbol{v} - F_{\nabla\cdot(\boldsymbol{v})} = 0, \tag{53}$$

where

$$\begin{aligned}
\boldsymbol{F}_{\nabla p} &= \nabla H^+(p^+ - p^-),\\
\boldsymbol{F}_{\nabla^2\boldsymbol{v}} &= \nabla H^+\cdot(\nabla\boldsymbol{v}^+ - \nabla\boldsymbol{v}^-) + \nabla\cdot(\nabla H^+(\boldsymbol{v}^+ - \boldsymbol{v}^-)),\\
\boldsymbol{F}_{\nabla\cdot(\boldsymbol{v}\boldsymbol{v})} &= \nabla H^+\cdot(\boldsymbol{v}^+\boldsymbol{v}^+ - \boldsymbol{v}^-\boldsymbol{v}^-),\\
\boldsymbol{F}_{\mathrm{d}\boldsymbol{v}/\mathrm{d}t} &= (\nabla H^+\cdot\dot{\boldsymbol{X}})(\boldsymbol{v}^+ - \boldsymbol{v}^-),\\
F_{\nabla\cdot\boldsymbol{v}} &= \nabla H^+\cdot(\boldsymbol{v}^+ - \boldsymbol{v}^-).
\end{aligned}$$

In the current case, we prescribe no-slip boundary conditions on the IB for both the exterior and interior solutions. As a result, $\boldsymbol{v}^+(\boldsymbol{X}(\boldsymbol{\xi},t),t) = \boldsymbol{v}^-(\boldsymbol{X}(\boldsymbol{\xi},t),t)$ for all $\boldsymbol{\xi}\in\Gamma$ and t, and we can simplify Eq. (52) to the standard continuous forcing

IB formulation,

$$\frac{\partial \boldsymbol{v}}{\partial t} + \nabla \cdot (\boldsymbol{v}\boldsymbol{v}) = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2 \boldsymbol{v} + \int_\Gamma \boldsymbol{f}(\boldsymbol{X}(\boldsymbol{\xi},t),t)\delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{\xi},t))\mathrm{d}S(\boldsymbol{\xi}), \quad \boldsymbol{x} \in \Omega, \tag{54}$$

$$\nabla \cdot \boldsymbol{v} = 0, \quad \boldsymbol{x} \in \Omega \tag{55}$$

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{v}_{\partial\Omega}(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega, \tag{56}$$

$$\int_\Omega \boldsymbol{v}(\boldsymbol{x},t)\delta(\boldsymbol{x} - \boldsymbol{X}(\boldsymbol{\xi},t))\mathrm{d}\boldsymbol{x} = \boldsymbol{v}_\Gamma(\boldsymbol{\xi},t), \quad \boldsymbol{\xi} \in \Gamma(t), \tag{57}$$

where $\boldsymbol{f}$ is the forcing density on Γ defined as

$$\boldsymbol{f}(\boldsymbol{X}(\boldsymbol{\xi},t),t) \coloneqq (p^+(\boldsymbol{X}(\boldsymbol{\xi}),t) - p^-(\boldsymbol{X}(\boldsymbol{\xi}),t))\mathbf{n}(\boldsymbol{\xi},t) + \frac{1}{\mathrm{Re}}(\boldsymbol{v}_\Gamma^{n+}(\boldsymbol{\xi},t) - \boldsymbol{v}_\Gamma^{n-}(\boldsymbol{\xi},t)), \tag{58}$$

with $\boldsymbol{v}_\Gamma^{n\pm}(\boldsymbol{\xi},t) \coloneqq \boldsymbol{n}(\boldsymbol{\xi},t) \cdot \nabla \boldsymbol{v}(\boldsymbol{X}(\boldsymbol{\xi}),t)$ the normal derivative of the exterior and interior velocity at the interface.

3.1. Discrete problem formulation for a composite solution

As in the Poisson problem, we derive the discrete formulation starting from the discrete composite velocity field $\overline{\mathbf{v}} = (\overline{\mathsf{v}}_x, \overline{\mathsf{v}}_y, \overline{\mathsf{v}}_z) \in \mathbb{R}^{\mathcal{F}}$ and pressure field $\overline{\mathsf{p}} \in \mathbb{R}^{\mathcal{C}}$, defined as

$$\overline{\mathbf{v}} \coloneqq \mathbf{H}_{\mathcal{F}}^+ \circ \mathbf{v}^+ + \mathbf{H}_{\mathcal{F}}^- \circ \mathbf{v}^- \tag{59}$$

$$\overline{\mathsf{p}} \coloneqq \mathsf{H}_{\mathcal{C}}^+ \circ \mathsf{p}^+ + \mathsf{H}_{\mathcal{C}}^- \circ \mathsf{p}^-, \tag{60}$$

where $\mathbf{v}^\pm$ and $\mathsf{p}^\pm$ are the discrete approximations to the exterior and interior velocity and pressure fields, respectively, and $\mathbf{H}_{\mathcal{F}}^\pm \coloneqq {}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\mathsf{H}_{\mathcal{C}}^\pm \in \mathbb{R}^{\mathcal{F}}$ is the interpolation of $\mathsf{H}_{\mathcal{C}}^\pm$ to each cell face.

To discretize the vector-valued momentum equation, we introduce the discrete Laplacian and gradient for cell face data, $\mathsf{L}_{\mathcal{F}} : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{\mathcal{F}}$ and $\mathsf{G}_{\mathcal{F}} : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{\mathcal{D}}$, where $\mathbb{R}^{\mathcal{D}}$ is the space for discrete, second-order tensor fields, such as the velocity gradient, whose components are defined on the following combination of cell locations:

$$\mathcal{D} = \begin{pmatrix} \mathcal{C} & \mathcal{E}_z & \mathcal{E}_y \\ \mathcal{E}_z & \mathcal{C} & \mathcal{E}_x \\ \mathcal{E}_y & \mathcal{E}_x & \mathcal{C} \end{pmatrix}. \tag{61}$$

We also define the divergence for second-order tensor data, $\mathsf{D}_{\mathcal{D}} : \mathbb{R}^{\mathcal{D}} \mapsto \mathbb{R}^{\mathcal{F}}$, and the interpolation and expansion/contraction from vectors to second-order tensors, ${}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}} : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{\mathcal{D}}$, and vice versa, ${}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}} : \mathbb{R}^{\mathcal{D}} \mapsto \mathbb{R}^{\mathcal{F}}$ (see Appendix A.1 for details). Lastly, we also define a space $\mathbb{R}^{\mathcal{T}}$ to hold second-order tensor data at the IB points.

We seek the solution for $\overline{\mathbf{v}}$ and $\overline{\mathsf{p}}$ such that the interior and exterior velocity and pressure fields satisfy the following spatially-discretized, non-dimensionalized, incompressible Navier-Stokes equations in their respective domains (assuming uniform properties in the entire domain),

$$\frac{\mathrm{d}\mathbf{v}^\pm}{\mathrm{d}t} + \mathsf{N}(\mathbf{v}^\pm) = -\mathsf{G}\mathsf{p}^\pm + \frac{1}{\mathrm{Re}}\mathsf{L}_{\mathcal{F}}\mathbf{v}^\pm + \mathbf{b}_1, \tag{62}$$

$$\mathsf{D}\mathbf{v}^\pm = \mathbf{b}_2, \tag{63}$$

where N is the discrete approximation to the convective term and $\mathbf{b}_1^\pm$ and $\mathbf{b}_2^\pm$ represent the boundary-condition terms that arises from discretizing the operators in the momentum and continuity equations, respectively, while enforcing the boundary conditions (50).

The corresponding equations for $\overline{\mathbf{v}}$ and $\overline{\mathsf{p}}$ can be found by first multiplying the Navier-Stokes equations for the interior and exterior fields by their respective indicator functions and summing them together,

$$\mathbf{H}_{\mathcal{F}}^+ \circ \left(\frac{\mathrm{d}\mathbf{v}^+}{\mathrm{d}t} + \mathsf{N}(\mathbf{v}^+)\right) + \mathbf{H}_{\mathcal{F}}^- \circ \left(\frac{\mathrm{d}\mathbf{v}^-}{\mathrm{d}t} + \mathsf{N}(\mathbf{v}^-)\right) = -\mathbf{H}_{\mathcal{F}}^+ \circ \mathsf{G}\mathsf{p}^+ - \mathbf{H}_{\mathcal{F}}^- \circ \mathsf{G}\mathsf{p}^- + \frac{1}{\mathrm{Re}}\left(\mathbf{H}_{\mathcal{F}}^+ \circ \mathsf{L}_{\mathcal{F}}\mathbf{v}^+ + \mathbf{H}_{\mathcal{F}}^- \circ \mathsf{L}_{\mathcal{F}}\mathbf{v}^-\right) + \mathbf{b}_1, \tag{64}$$

$$\mathsf{H}_{\mathcal{C}}^+ \circ \mathsf{D}\mathbf{v}^+ + \mathsf{H}_{\mathcal{C}}^- \circ \mathsf{D}\mathbf{v}^- = \mathbf{b}_2. \tag{65}$$

Then, we replace the matching exterior and interior terms by the involved operator acting on the composite fields minus a "forcing" term from the relations (A.22), (A.25), (A.31), (A.32), and (A.35) in Appendix A.4, resulting in the discrete analogue of (52)-(53):

$$\frac{\mathrm{d}\overline{\mathbf{v}}}{\mathrm{d}t} + \mathsf{N}(\overline{\mathbf{v}}) - \mathbf{F}_{\mathrm{d}\overline{\mathbf{v}}/\mathrm{d}t} - \mathbf{F}_{\mathsf{N}(\overline{\mathbf{v}})} = -\mathsf{G}\overline{\mathsf{p}} + \frac{1}{\mathrm{Re}}\mathsf{L}_{\mathcal{F}}\overline{\mathbf{v}} + \mathsf{F}_{\mathsf{G}\overline{\mathsf{p}}} - \frac{1}{\mathrm{Re}}\mathbf{F}_{\mathsf{L}_{\mathcal{F}}\overline{\mathbf{v}}} + \mathbf{b}_1, \tag{66}$$

$$\mathsf{D}\overline{\mathbf{v}} - \mathsf{F}_{\mathsf{D}\overline{\mathbf{v}}} = \mathsf{b}_2, \tag{67}$$

where

$$\begin{aligned}
\mathbf{F}_{\mathsf{G}\overline{\mathsf{p}}} &= \mathsf{G}\mathsf{H}_C^+ \circ {}^{\mathcal{F}}\mathsf{I}_C(\mathsf{p}^+ - \mathsf{p}^-),\\
\mathbf{F}_{\mathsf{L}_{\mathcal{F}}\overline{\mathbf{v}}} &= {}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\Big((\mathsf{I}_{\mathcal{F}}\mathsf{G}\mathsf{H}_C^+)^{\top} \circ (\mathsf{G}_{\mathcal{F}}\mathbf{v}^+ - \mathsf{G}_{\mathcal{F}}\mathbf{v}^-)\Big) + \mathsf{D}_{\mathcal{D}}\Big((\mathsf{I}_{\mathcal{F}}\mathsf{G}\mathsf{H}_C^+)^{\top} \circ \mathsf{I}_{\mathcal{F}}(\mathbf{v}^+ - \mathbf{v}^-)\Big),\\
\mathbf{F}_{\mathsf{N}(\overline{\mathbf{v}})} &= {}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\Big(\mathsf{G}_{\mathcal{F}}\mathbf{H}_{\mathcal{F}}^+ \circ \Big((\mathsf{I}_{\mathcal{F}}\mathbf{v}^+)^{\top} \circ \mathsf{I}_{\mathcal{F}}\mathbf{v}^+ - (\mathsf{I}_{\mathcal{F}}\mathbf{v}^-)^{\top} \circ \mathsf{I}_{\mathcal{F}}\mathbf{v}^-\Big)\Big)\\
&\quad - \mathsf{D}_{\mathcal{D}}\Big(\mathbf{H}_{\mathcal{D}}^+ \circ \mathbf{H}_{\mathcal{D}}^- \circ (\mathsf{I}_{\mathcal{F}}(\mathbf{v}^+ - \mathbf{v}^-))^{\top} \circ \mathsf{I}_{\mathcal{F}}(\mathbf{v}^+ - \mathbf{v}^-)\Big) + O(\Delta\boldsymbol{x}^2),\\
\mathbf{F}_{\mathrm{d}\overline{\mathbf{v}}/\mathrm{d}t} &= \frac{\mathrm{d}\mathbf{H}_{\mathcal{F}}^+}{\mathrm{d}t} \circ (\mathbf{v}^+ - \mathbf{v}^-),\\
\mathbf{F}_{\mathsf{D}\overline{\mathbf{v}}} &= {}^{C}\mathsf{I}_{\mathcal{F}}(\mathsf{G}\mathsf{H}_C^+ \circ (\mathbf{v}^+ - \mathbf{v}^-)).
\end{aligned}$$

In each of the preceding terms, except $\mathbf{F}_{\mathrm{d}\overline{\mathbf{v}}/\mathrm{d}t}$, we replace $\mathsf{G}\mathsf{H}_C^+$ with $\mathsf{R}_{\mathcal{F}}\mathbf{n}$.

3.2. Regularization operations

Similar to our treatment of the Poisson problem, we approximate the variation of the pressure and velocity within the support of a discrete DDF by a truncated Taylor series expansion about the discrete surface point at which the DDF is centered. For the pressure we use only the zeroth-order term of (A.36) and for the velocity and its gradient we use (A.38) and (A.39). We will make use of the notation $\mathbf{v}_\Gamma^{\pm}$, $\mathbf{v}_\Gamma^{n\pm}$, $\mathbf{v}_\Gamma^{t\pm}$, and $\mathbf{v}_\Gamma^{b\pm}$ to denote the value and normal and tangential derivatives of the exterior and interior velocity fields at the IB points, and we will use $\mathsf{p}_\Gamma^{\pm}$ to denote the exterior and interior pressure at the IB points. Because $\mathbf{v}_\Gamma^+ = \mathbf{v}_\Gamma^-$, we have that $[\mathbf{v}]_\Gamma = \mathbf{0}$ and also $[\mathbf{v}^t]_\Gamma = \mathbf{0}$ and $[\mathbf{v}^b]_\Gamma = \mathbf{0}$. Consequently, we can approximate the expression in the outermost parentheses of the first term of $\mathbf{F}_{\mathsf{L}_{\mathcal{F}}\overline{\mathbf{v}}}$ as

$$\begin{aligned}
&\Big\{(\mathsf{I}_{\mathcal{F}}\mathsf{R}_{\mathcal{F}}\mathbf{n})^{\top} \circ (\mathsf{G}_{\mathcal{F}}\mathbf{v}^+ - \mathsf{G}_{\mathcal{F}}\mathbf{v}^-)\Big\}_{i,j,k}\\
&\quad\approx \sum_l \begin{pmatrix}
\{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{n}_x\}_l\{[\mathrm{v}_x^n]_\Gamma\}_l\{\mathrm{n}_x\}_l & \{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{n}_y\}_l\{[\mathrm{v}_x^n]_\Gamma\}_l\{\mathrm{n}_y\}_l & \{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{n}_z\}_l\{[\mathrm{v}_x^n]_\Gamma\}_l\{\mathrm{n}_z\}_l\\
\{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{n}_x\}_l\{[\mathrm{v}_y^n]_\Gamma\}_l\{\mathrm{n}_x\}_l & \{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{n}_y\}_l\{[\mathrm{v}_y^n]_\Gamma\}_l\{\mathrm{n}_y\}_l & \{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{n}_z\}_l\{[\mathrm{v}_y^n]_\Gamma\}_l\{\mathrm{n}_z\}_l\\
\{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{n}_x\}_l\{[\mathrm{v}_z^n]_\Gamma\}_l\{\mathrm{n}_x\}_l & \{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{n}_y\}_l\{[\mathrm{v}_z^n]_\Gamma\}_l\{\mathrm{n}_y\}_l & \{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{n}_z\}_l\{[\mathrm{v}_z^n]_\Gamma\}_l\{\mathrm{n}_z\}_l
\end{pmatrix}\{\mathrm{S}\}_l\\
&\quad= \Big\{\mathsf{R}_{(\mathsf{I}\mathcal{F})^{\top}}\mathsf{O}_{\mathbf{n}\circ\mathbf{n}}([\mathbf{v}^n]_\Gamma)\Big\}_{i,j,k},
\end{aligned} \tag{68}$$

where we used the shorthand $\mathsf{R}_{(\mathsf{I}\mathcal{F})^{\top}} : \mathbb{R}^{\mathcal{T}} \mapsto \mathbb{R}^{\mathcal{D}}$ defined in (A.41) and where $\mathsf{O}_{\mathbf{v}_1} : \mathbb{R}^{\mathcal{V}} \mapsto \mathbb{R}^{\mathcal{T}}$ is an operator that acts on vector-valued point data and produces the outer product with $\mathbf{v}_1$,

$$\{\mathsf{O}_{\mathbf{v}_1}(\mathbf{v}_2)\}_l \coloneqq \{\mathbf{v}_1\}_l\{\mathbf{v}_2\}_l^{\top}. \tag{69}$$

Similarly, if we use the shorthand $\mathsf{R}_{(\mathsf{I}\mathcal{F})^{\top},1n} \in \mathbb{R}^{\mathcal{T}} \mapsto \mathbb{R}^{\mathcal{D}}$ defined in (A.42), we can approximate the expression in the outermost parentheses of the second term of $\mathbf{F}_{\mathsf{L}_{\mathcal{F}}\overline{\mathbf{v}}}$ as

$$(\mathsf{I}_{\mathcal{F}}\mathsf{R}_{\mathcal{F}}\mathbf{n})^{\top} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}(\mathbf{v}^+ - \mathbf{v}^-) \approx \mathsf{R}_{(\mathsf{I}\mathcal{F})^{\top},1n}\mathsf{O}_{\mathbf{n}}([\mathbf{v}^n]_\Gamma), \tag{70}$$

and we approximate $\mathbf{F}_{\mathsf{D}\overline{\mathbf{v}}}$ and $\mathbf{F}_{\mathsf{G}\overline{\mathsf{p}}}$ as

$$\mathsf{R}_{\mathcal{F}}\mathbf{n} \circ (\mathbf{v}^+ - \mathbf{v}^-) \approx \mathsf{R}_{\mathcal{F},1n}(\mathbf{n} \circ [\mathbf{v}^n]_\Gamma), \tag{71}$$

$$\mathsf{R}_{\mathcal{F}}\mathbf{n} \circ {}^{\mathcal{F}}\mathsf{I}_C(\mathsf{p}^+ - \mathsf{p}^-) \approx \mathsf{R}_{\mathcal{F}}(\mathbf{n} \circ [\mathrm{p}]_\Gamma). \tag{72}$$

The term $\mathsf{F}_{\mathsf{N}(\overline{\mathsf{v}})}$—which is only nonzero within the support of the discrete DDF—can be ignored since its magnitude inside this support scales as $O(\Delta \boldsymbol{x})$ while the other terms scale as $O(1/\Delta \boldsymbol{x})$. This can be seen by considering that the Riemann sum (21) with the discrete DDF (3) scales as $O(1/\Delta \boldsymbol{x})$, while the Taylor series of the velocity fields scale as $O(\Delta \boldsymbol{x})$ within the DDF support. A divergence or gradient operation increases the scaling by another factor of $\Delta \boldsymbol{x}$. Since the first term of $\mathsf{F}_{\mathsf{N}(\overline{\mathsf{v}})}$ involves the product of two velocity fields, which scales as $O(\Delta \boldsymbol{x}^2)$, the entire first term involving $\mathsf{G}_{\mathcal{F}}\mathbf{H}^+_{\mathcal{F}}$ scales as $O(\Delta \boldsymbol{x})$. Similarly, one can show that the second term of $\mathsf{F}_{\mathsf{N}(\overline{\mathsf{v}})}$ scales as $O(\Delta \boldsymbol{x})$.

For the time derivative of the discrete indicator function, which appears in the term $\mathsf{F}_{\mathrm{d}\overline{\mathbf{v}}/\mathrm{d}t}$, we directly discretize $\nabla H^+ \cdot \dot{\boldsymbol{X}} = \int (\boldsymbol{n} \cdot \dot{\boldsymbol{X}}) \delta(\boldsymbol{x} - \boldsymbol{X}) \mathrm{d}S(\boldsymbol{\xi})$ [27, 3] as $\mathsf{R}_{\mathcal{F}}(\dot{\mathbf{X}}_n)$ and obtain

$$\frac{\mathrm{d}\mathbf{H}^+_{\mathcal{F}}}{\mathrm{d}t} \circ (\mathbf{v}^+ - \mathbf{v}^-) \approx -\mathsf{R}_{\mathcal{F},1n}(\dot{\mathbf{X}}_n \circ [\mathbf{v}^n]_\Gamma), \tag{73}$$

where $\{\dot{\mathbf{X}}_n\}_l \coloneqq \{\mathbf{n}\}_l \cdot \{\dot{\mathbf{X}}\}_l$ with $\dot{\mathbf{X}}$ the velocity of the l-th IB point. Alternatively, one can compute $\mathrm{d}\mathbf{H}^+_{\mathcal{F}}/\mathrm{d}t$ more accurately as:

$$\begin{aligned}
\frac{\mathrm{d}\mathbf{H}^+_{\mathcal{F}}}{\mathrm{d}t} &= {}^{\mathcal{F}}\mathsf{I}_C \frac{\mathrm{d}}{\mathrm{d}t} \mathsf{L}^{-1} \mathsf{D} \mathsf{R}_{\mathcal{F}} \mathbf{n} \\
&= {}^{\mathcal{F}}\mathsf{I}_C \mathsf{L}^{-1} \mathsf{D} \sum_l \begin{pmatrix} \{\mathrm{n}_x\}_l \mathrm{d}(\mathsf{d}_{\mathcal{F}_x,l})/\mathrm{d}t + \mathsf{d}_{\mathcal{F}_x,l} \mathrm{d}\{\mathrm{n}_x\}_l/\mathrm{d}t \\ \{\mathrm{n}_y\}_l \mathrm{d}(\mathsf{d}_{\mathcal{F}_y,l})/\mathrm{d}t + \mathsf{d}_{\mathcal{F}_y,l} \mathrm{d}\{\mathrm{n}_y\}_l/\mathrm{d}t \\ \{\mathrm{n}_z\}_l \mathrm{d}(\mathsf{d}_{\mathcal{F}_z,l})/\mathrm{d}t + \mathsf{d}_{\mathcal{F}_z,l} \mathrm{d}\{\mathrm{n}_z\}_l/\mathrm{d}t \end{pmatrix} \{\mathrm{S}\}_l \\
&= -{}^{\mathcal{F}}\mathsf{I}_C \mathsf{L}^{-1} \mathsf{D} \sum_l \begin{pmatrix} \{\mathrm{n}_x\}_l \nabla \delta_{\Delta \boldsymbol{x}}(\mathbf{x}_{\mathcal{F}_x} - \{\mathbf{X}\}_l) \cdot \{\dot{\mathbf{X}}\}_l + \mathsf{d}_{\mathcal{F}_x,l} \mathrm{d}\{\mathrm{n}_x\}_l/\mathrm{d}t \\ \{\mathrm{n}_y\}_l \nabla \delta_{\Delta \boldsymbol{x}}(\mathbf{x}_{\mathcal{F}_y} - \{\mathbf{X}\}_l) \cdot \{\dot{\mathbf{X}}\}_l + \mathsf{d}_{\mathcal{F}_y,l} \mathrm{d}\{\mathrm{n}_y\}_l/\mathrm{d}t \\ \{\mathrm{n}_z\}_l \nabla \delta_{\Delta \boldsymbol{x}}(\mathbf{x}_{\mathcal{F}_z} - \{\mathbf{X}\}_l) \cdot \{\dot{\mathbf{X}}\}_l + \mathsf{d}_{\mathcal{F}_z,l} \mathrm{d}\{\mathrm{n}_z\}_l/\mathrm{d}t \end{pmatrix} \{\mathrm{S}\}_l.
\end{aligned} \tag{74}$$

Using the approximations (68), (70), (71), (72), and (73), we can formulate the final versions of the spatially-discretized momentum and continuity equations for the composite fields:

$$\begin{aligned}
&\frac{\mathrm{d}\overline{\mathbf{v}}}{\mathrm{d}t} + \mathsf{N}(\overline{\mathbf{v}}) + \mathsf{R}_{\mathcal{F},1n}(\dot{\mathbf{X}}_n \circ [\mathbf{v}^n]_\Gamma) = \\
&\qquad - \mathsf{G}\overline{\mathsf{p}} + \frac{1}{\mathrm{Re}} \mathsf{L}_{\mathcal{F}} \overline{\mathbf{v}} + \mathsf{R}_{\mathcal{F}}(\mathbf{n} \circ [\mathrm{p}]_\Gamma) - \frac{1}{\mathrm{Re}} \Big({}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}} \mathsf{R}_{(\mathsf{I}\mathcal{F})^\top} \mathsf{O}_{\mathbf{n} \circ \mathbf{n}}([\mathbf{v}^n]_\Gamma) + \mathsf{D}_{\mathcal{D}} \mathsf{R}_{(\mathsf{I}\mathcal{F})^\top,1n} \mathsf{O}_{\mathbf{n}}([\mathbf{v}^n]_\Gamma) \Big) + \mathbf{b}_1,
\end{aligned} \tag{75}$$

$$\mathsf{D}\overline{\mathbf{v}} = {}^{C}\mathsf{I}_{\mathcal{F}} \mathsf{R}_{\mathcal{F},1n}(\mathbf{n} \circ [\mathbf{v}^n]_\Gamma) + \mathbf{b}_2. \tag{76}$$

3.3. Interpolation of the velocity and pressure to the immersed boundary

We also use the Taylor series construction (A.38) for $\mathbf{v}$ near the IB points to improve the accuracy of the interpolation operator (22) for vector-valued data. In our case that $\mathbf{v}^{\pm}_\Gamma = \mathbf{v}_\Gamma$, we follow the approach equivalent to (32), but now for vector-valued data, to obtain the following second-order accurate interpolation for the composite solution (provided we use a DDF that satisfies the first-order discrete moment condition):

$$\mathsf{E}_{\mathcal{F}} \overline{\mathbf{v}} - [\mathbf{v}^n]_\Gamma \circ \mathsf{E}_{\mathcal{F},1n} \mathbf{H}^+_{\mathcal{F}} \approx \mathbf{v}_\Gamma, \tag{77}$$

where we introduced the normal-distance-weighted interpolation operation $\mathsf{E}_{\mathcal{F},1n} : \mathbb{R}^{\mathcal{V}} \mapsto \mathbb{R}^{\mathcal{F}}$, defined as:

$$\{\mathsf{E}_{\mathcal{F},1n} \mathbf{v}\}_l \coloneqq \Delta x \Delta y \Delta z \sum_{i,j,k} \begin{pmatrix} \{\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k} \mathbf{n}_l \cdot (\{\mathbf{x}_{\mathcal{F}_x}\}_{i,j,k} - \{\mathbf{X}\}_l) \{\mathsf{v}_x\}_{i,j,k} \\ \{\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k} \mathbf{n}_l \cdot (\{\mathbf{x}_{\mathcal{F}_y}\}_{i,j,k} - \{\mathbf{X}\}_l) \{\mathsf{v}_y\}_{i,j,k} \\ \{\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k} \mathbf{n}_l \cdot (\{\mathbf{x}_{\mathcal{F}_z}\}_{i,j,k} - \{\mathbf{X}\}_l) \{\mathsf{v}_z\}_{i,j,k} \end{pmatrix}. \tag{78}$$

The discrete momentum equation (75) contains the jump of the pressure $[\mathrm{p}]_\Gamma$ across the interface in addition to the composite pressure field $\overline{\mathsf{p}}$. We can find a relation between both by using the normal-distance-weighted interpolation

operation, the relation (33) for DDFs that satisfy the first-order discrete moment condition, and the zeroth-order term of the Taylor series expansion (A.36) for $\mathsf{p}^{\pm}$:

$$\begin{aligned}
\{\mathsf{E}_{C,1n}\overline{\mathsf{p}}\}_l &= \Delta x\Delta y\Delta z \sum_{i,j,k} \{\mathsf{d}_{C,l}\}_{i,j,k}\Big(\{\mathbf{n}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\Big)\Big(\{\mathsf{H}_C^+ \circ \mathsf{p}^+\}_{i,j,k} + \{\mathsf{H}_C^- \circ \mathsf{p}^-\}_{i,j,k}\Big) \\
&= \Delta x\Delta y\Delta z \sum_{i,j,k} \{\mathsf{d}_{C,l}\}_{i,j,k}\Big(\{\mathbf{n}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\Big)\Big(\{\mathsf{H}_C^+\}_{i,j,k}\{\mathsf{p}_\Gamma^+\}_l + \{1 - \mathsf{H}_C^+\}_{i,j,k}\{\mathsf{p}_\Gamma^-\}_l\Big) \\
&= \{[\mathsf{p}]_\Gamma\}_l \Delta x\Delta y\Delta z \sum_{i,j,k} \{\mathsf{d}_{C,l}\}_{i,j,k}\Big(\{\mathbf{n}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\Big)\{\mathsf{H}_C^+\}_{i,j,k} \\
&= \{[\mathsf{p}]_\Gamma \circ \mathsf{E}_{C,1n}\mathsf{H}_C^+\}_l. \qquad (79)
\end{aligned}$$

We also define a modified normal-distance-weighted interpolation operation $\tilde{\mathsf{E}}_{C,1n}$ that removes the average value from the final result:

$$\tilde{\mathsf{E}}_{C,1n}\overline{\mathsf{p}} = \mathsf{E}_{C,1n}\overline{\mathsf{p}} - \sum_l \{\mathsf{E}_{C,1n}\overline{\mathsf{p}}\}_l/N_l. \qquad (80)$$

3.4. Projection-based approach to solving the constrained system

For simplicity, we discretize the discrete momentum equation (75) in time using a fully-explicit time-stepping scheme and we define the right-hand-side vector $\mathbf{r} \in \mathcal{F}$,

$$\mathbf{r} = \overline{\mathbf{v}} - K\Delta t\mathsf{N}(\overline{\mathbf{v}}) + \frac{K\Delta t}{\mathrm{Re}}\mathsf{L}_{\mathcal{F}}\overline{\mathsf{v}}, \qquad (81)$$

where Δt is the time-step size and $K \in \mathbb{R}$ is a coefficient that depends on the chosen time-stepping scheme. Then, we can formulate the discrete Navier-Stokes system that we have to solve to advance the solution by one stage of the timestepping scheme. This system consists of the temporally-discretized momentum equation (75), constrained by the continuity equation (76), the IB boundary condition (77), and the relation between the pressure field and pressure jump (79), which we can formulate as a saddle point system,

$$\begin{bmatrix} \mathsf{I} & \Delta t\mathsf{G} & K\Delta t\mathsf{R}_{\mathcal{F},1n}\mathrm{diag}(\dot{\mathbf{X}}_n) + \frac{K\Delta t}{\mathrm{Re}}({}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\mathsf{R}_{(\mathsf{I}\mathcal{F})^\top}\mathsf{O}_{\mathbf{n}\circ\mathbf{n}} + \mathsf{D}_{\mathcal{D}}\mathsf{R}_{(\mathsf{I}\mathcal{F})^\top,1n}\mathsf{O}_{\mathbf{n}}) & -K\Delta t\mathsf{R}_{\mathcal{F}}\mathrm{diag}(\mathbf{n}) \\ \mathsf{D} & 0 & -{}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\mathsf{R}_{\mathcal{F},1n} & 0 \\ \mathsf{E}_{\mathcal{F}} & 0 & -\mathrm{diag}(\mathsf{E}_{\mathcal{F},1n}\mathbf{H}_{\mathcal{F}}^+) & 0 \\ 0 & \tilde{\mathsf{E}}_{C,1n} & 0 & -\mathrm{diag}(\mathsf{E}_{C,1n}\mathsf{H}_C^+) \end{bmatrix} \begin{pmatrix} \overline{\mathbf{v}} \\ \overline{\mathsf{p}} \\ [\mathbf{v}_n]_\Gamma \\ [\mathsf{p}]_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathsf{b} \\ \mathbf{v}_\Gamma \\ 0 \end{pmatrix}, \qquad (82)$$

where $\mathbf{r}$ is constructed using the solution values from the previous timestepping stage. For comparison, we also formulate the original IBPM system by Taira and Colonius [12], which is a direct discretization of (54)-(57):

$$\begin{bmatrix} \mathsf{I} & \Delta t\mathsf{G} & K\Delta t\mathsf{R}_{\mathcal{F}} \\ \mathsf{D} & 0 & 0 \\ \mathsf{E}_{\mathcal{F}} & 0 & 0 \end{bmatrix} \begin{pmatrix} \overline{\mathbf{v}} \\ \overline{\mathsf{p}} \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathsf{b} \\ \mathbf{v}_\Gamma \end{pmatrix}, \qquad (83)$$

where $\mathbf{f}$ is the IB forcing strength, which is the discrete analogue of (58). One key difference between the new and the original IBPM system is the presence of the diagonal matrices in the lower-right blocks. Similar to the Poisson problem, the presence of these matrices has the effect of removing the ill-conditioning associated with the original IBPM system.

To solve the proposed system (82), we first define the block matrices,

$$B_{1,1}^\top = \Big[K\Delta t\mathsf{R}_{\mathcal{F},1n}\mathrm{diag}(\dot{\mathbf{X}}_n) + \frac{K\Delta t}{\mathrm{Re}}({}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\mathsf{R}_{(\mathsf{I}\mathcal{F})^\top}\mathsf{O}_{\mathbf{n}\circ\mathbf{n}} + \mathsf{D}_{\mathcal{D}}\mathsf{R}_{(\mathsf{I}\mathcal{F})^\top,1n}\mathsf{O}_{\mathbf{n}}) \quad -K\Delta t\mathsf{R}_{\mathcal{F}}\mathrm{diag}(\mathbf{n})\Big], \qquad (84)$$

$$B_{1,2}^\top = \Big[-{}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\mathsf{R}_{\mathcal{F},1n}\mathrm{diag}(\mathbf{n}) \quad 0\Big], \qquad (85)$$

and then formulate the solution procedure based on Schur complement reduction method,

$$\Delta t \mathsf{L}\overline{\mathsf{p}}^* = \mathsf{D}\mathbf{r} - \mathsf{b} \tag{86}$$

$$\overline{\mathbf{v}}^* = \mathbf{r} - \Delta t \mathsf{G}\overline{\mathsf{p}}^* \tag{87}$$

$$\tag{88}$$

$$S\begin{pmatrix}[\mathbf{v}_n]_\Gamma\\ [\mathsf{p}]_\Gamma\end{pmatrix} = \begin{pmatrix}\mathbf{v}_\Gamma\\ 0\end{pmatrix} - \begin{bmatrix}\mathsf{E}_{\mathcal{F}} & 0\\ 0 & \tilde{\mathsf{E}}_{C,1n}\end{bmatrix}\begin{pmatrix}\overline{\mathbf{v}}^*\\ \overline{\mathsf{p}}^*\end{pmatrix} \tag{89}$$

$$\overline{\mathsf{p}} = \overline{\mathsf{p}}^* - \frac{1}{\Delta t}\mathsf{L}^{-1}(\mathsf{D}B_{1,1}^\top - B_{1,2}^\top)\begin{pmatrix}[\mathbf{v}_n]_\Gamma\\ [\mathsf{p}]_\Gamma\end{pmatrix} \tag{90}$$

$$\overline{\mathbf{v}} = \mathbf{r} - B_{1,1}^\top\begin{pmatrix}[\mathbf{v}_n]_\Gamma\\ [\mathsf{p}]_\Gamma\end{pmatrix} - \Delta t \mathsf{G}\overline{\mathsf{p}} \tag{91}$$

where S is the Schur complement of the system:

$$S = \begin{bmatrix}-\mathsf{E}_{\mathcal{F}}B_{1,1}^\top + \mathsf{E}_{\mathcal{F}}\mathsf{G}\mathsf{L}^{-1}(\mathsf{D}B_{1,1}^\top - B_{1,2}^\top)\\ -\tilde{\mathsf{E}}_{C,1n}\mathsf{L}^{-1}(\mathsf{D}B_{1,1}^\top - B_{1,2}^\top)\end{bmatrix} + \begin{bmatrix}-\mathrm{diag}(\mathsf{E}_{\mathcal{F},1n}\mathbf{H}_{\mathcal{F}}^+) & 0\\ 0 & -\mathrm{diag}(\mathsf{E}_{C,1n}\mathsf{H}_C^+)\end{bmatrix}. \tag{92}$$

The solution procedure (87)-(91) involves several steps that require the solution of Poisson equations. In our implementation, these are solved using a fast Fourier transform (FFT)–based method with the zero Fourier mode set to zero. The procedure also requires the solution of the linear system (89). For moving bodies or problems with a large number of degrees of freedom, this system is most efficiently solved using an iterative method, such as a biconjugate gradient algorithm. Alternatively, the inverse of S may be computed directly using, for example, an LU factorization. However, this approach is more computationally expensive and is therefore only practical for small systems with stationary immersed boundaries.

In our system (82), we make use of the pressure interpolation (80) that produces a result with a zero spatial mean. This choice allows the iterative solver to converge to a unique solution, as the solution for $[\mathsf{p}]_\Gamma$ is only defined up to an additive constant. When directly inverting S, one also has to replace one row per closed IB curve in the final block row of (82)—specifically, Eq. (79) for one point on each closed IB curve—by an equation that sets the average of $[\mathsf{p}]_\Gamma$ to zero (per closed IB curve). For example, in case there is a single closed curve, we can use

$$\sum_l \{[\mathsf{p}]_\Gamma\}_l / N_l = 0. \tag{93}$$

3.5. Results for a 2D cylindrical Couette flow problem

To assess the accuracy and conditioning of the proposed method, we consider the two-dimensional Navier-Stokes problem of the circular Couette flow between two concentric cylinders with radii $R_1 = 0.5$ and $R_2 = 1$, shown in the left panel of Fig. 7. The inner cylinder Γ_1 rotates with a constant angular velocity ω, while the outer cylinder Γ_2 remains stationary. The Reynolds number based on the velocity and diameter of the inner cylinder is $\mathrm{Re} = 2\omega R_1^2/\nu = 0.225$. At this low Reynolds number, the flow remains a laminar, azimuthal Couette flow, which varies only in the radial direction. The dimensional, analytical solution to this problem is

$$\boldsymbol{v}_\theta(r) = \begin{cases}\omega r, & r \le R_1,\\ \omega\dfrac{R_1^2/r - r(R_1/R_2)^2}{1-(R_1/R_2)^2}, & R_1 < r \le R_2,\\ 0, & r > R_2,\end{cases} \tag{94}$$

where $\boldsymbol{v}_\theta$ is the azimuthal velocity. The analytical normal-derivative jumps over the inner and outer cylinders of the azimuthal velocity are

$$[\boldsymbol{v}_\theta^n]_{\Gamma_1} = \frac{2\omega}{1-(R_1/R_2)^2}, \quad [\boldsymbol{v}_\theta^n]_{\Gamma_2} = \omega\frac{1+(R_1/R_2)^2}{1-(R_1/R_2)^2}. \tag{95}$$

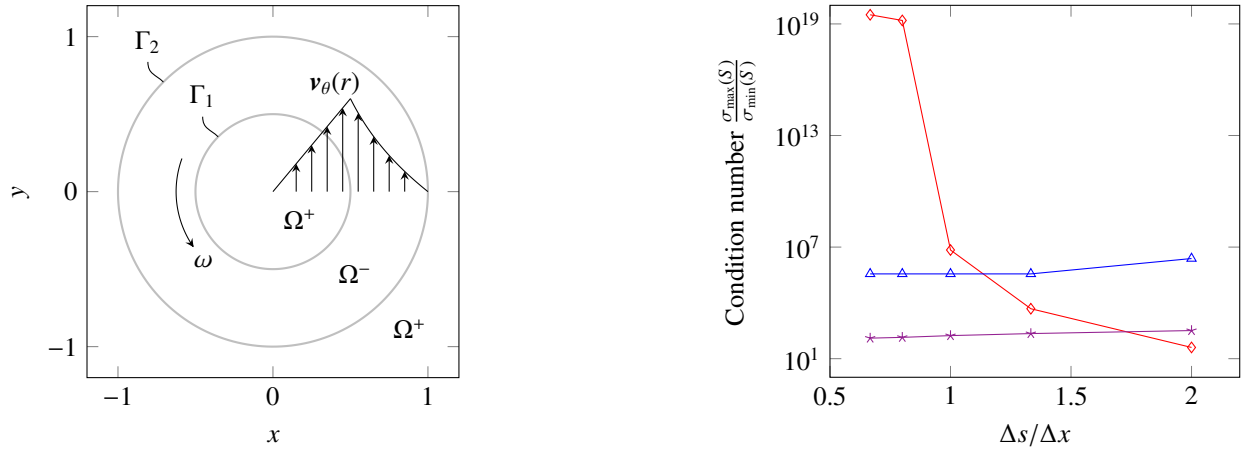

Figure 7: (Left) Diagram of the circular Couette flow problem. (Right) Condition number of the Schur complement S versus the surface-to-grid spacing ratio for the proposed method (blue, triangles), the hybrid method (purple, stars), and the original IBPM (red, diamonds) using grid cell size $\Delta x/R_1 = 0.167$.

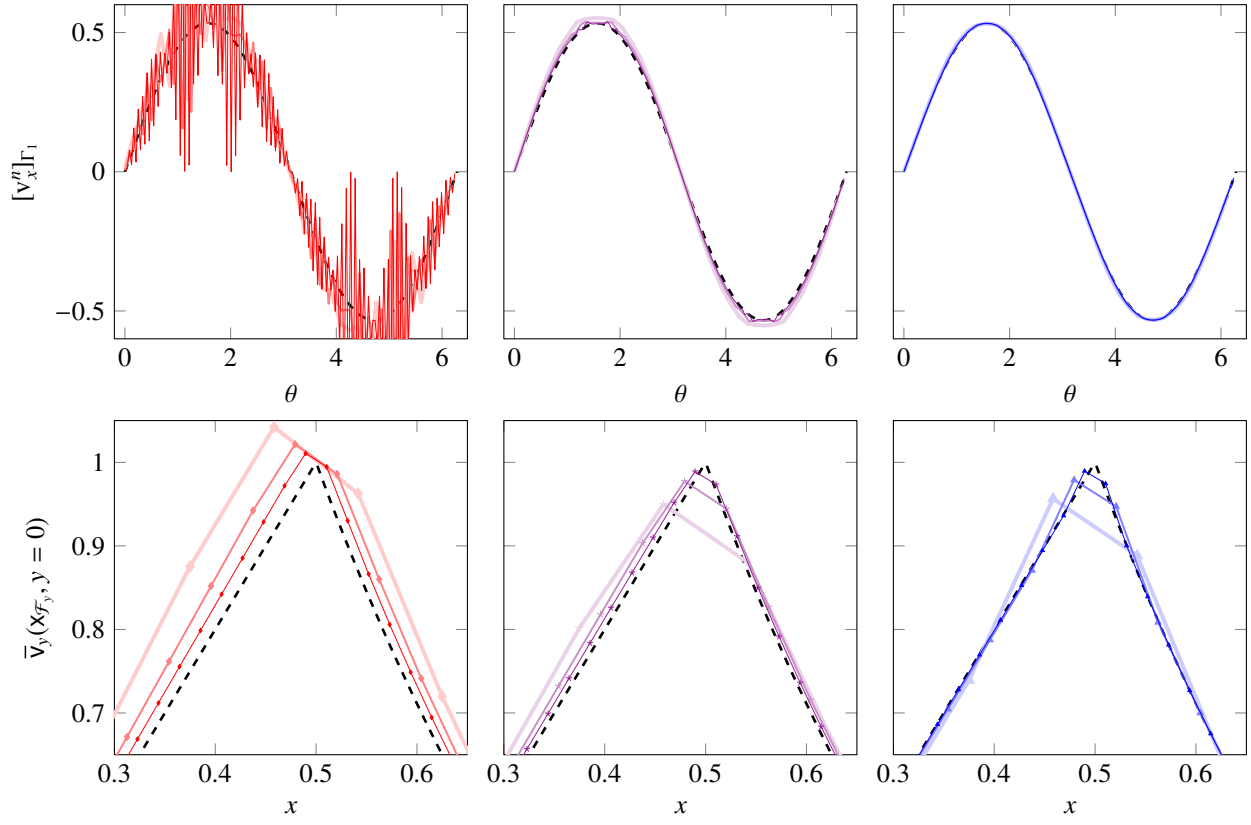

Figure 8: Comparison of the numerical forcing solution (top row) and vertical centerline velocity solution near the inner cylinder (bottom row) from the original method (red, diamonds), the solution from the hybrid method (purple, stars), and the proposed method (blue, triangles) with the exact solution (black, dashed) on the line along $y = 0$ near the interface for $\Delta s/\Delta x \approx 1$ and three different grid cell sizes Δx: 0.25 (lightest, thick), 0.125, and 0.0625 (darkest, thin).

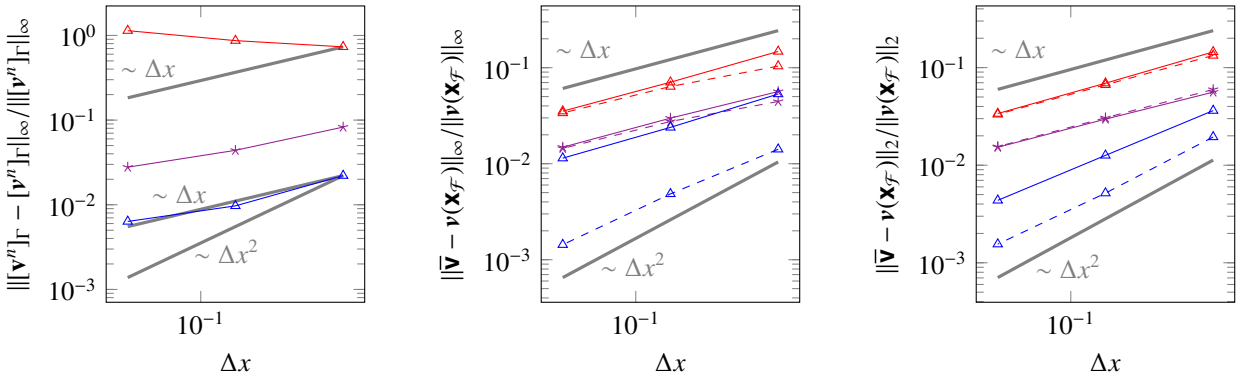

Figure 9: (Left) Infinity norm of the forcing strength error, and (center) infinity norm and (right) 2-norm of the velocity solution error with respect to their analytical values for the circular Couette flow example for the original method (red, diamonds), the hybrid method (purple, stars), and the proposed method (blue, triangles). The velocity error is computed over all cell centers (solid lines) and over all cell centers excluding those within three grid spacings from the interface (dashed lines). First- and second-order error scales are also shown (gray, solid).

We discretize the problem on the finite domain $\Omega = \left\{(x, y) \in \mathbb{R}^2 \;\middle|\; -1.33 \le x \le 1.33,\ -1.33 \le y \le 1.33\right\}$ uniformly using several grid sizes and body-to-grid spacing ratios, and we compare the resulting solutions obtained after time-stepping the system from zero initial conditions until steady-state using the original IB system (83), the proposed IB system (82), and a system consisting of the momentum and continuity equations of the original system (83) and the interface constraints (77) and (79) of the proposed system,

$$\begin{bmatrix} \mathsf{I} & \Delta t\mathsf{G} & K\Delta t\mathsf{R}_\mathcal{F} & -K\Delta t\mathsf{R}_\mathcal{F}\mathrm{diag}(\mathbf{n}) \\ \mathsf{D} & 0 & -{}^C\mathsf{I}_\mathcal{F}\mathsf{R}_{\mathcal{F},1n} & 0 \\ \mathsf{E}_\mathcal{F} & 0 & -\mathrm{diag}(\mathsf{E}_{\mathcal{F},1n}\mathbf{H}^+_\mathcal{F}) & 0 \\ 0 & \tilde{\mathsf{E}}_{C,1n} & 0 & -\mathrm{diag}(\mathsf{E}_{C,1n}\mathsf{H}^+_C) \end{bmatrix} \begin{pmatrix} \bar{\mathbf{v}} \\ \bar{\mathsf{p}} \\ [\mathbf{v}_n]_\Gamma \\ [\mathsf{p}]_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathsf{b} \\ \mathbf{v}_\Gamma \\ 0 \end{pmatrix}, \tag{96}$$

which we call the *hybrid* method and solve with the Schur complement reduction method, similar to the other systems. The right panel of Fig. 7 shows that, in terms of the Schur complement condition number, the original IB system rapidly becomes ill-conditioned for $\Delta s/\Delta x$ values near 1. In contrast, the proposed method and the hybrid system remain well-conditioned due to the introduction of the new constraint equations. As a consequence, the computed normal-derivative jumps—which are equivalent to the original IB forcing, since the pressure jump is zero for all methods—can become extremely noisy for the original method, while remaining relatively smooth for the proposed method and the hybrid system. This behavior is illustrated, for example, by the normal-derivative jump on the inner cylinder for the x-component of the velocity at $\Delta s/\Delta x \approx 1$, shown in the top row of Fig. 8.

The bottom row of Fig. 8 compares the solutions obtained with the three methods in the vicinity of the inner cylinder for three different grid sizes at $\Delta s/\Delta x \approx 1$. As in the Poisson problem, the original IB method exhibits a large error near the interface due to its first-order accurate interpolation scheme. In contrast, the hybrid and proposed methods show a significant reduction in error as a result of their improved interpolation schemes. Moreover, owing to the modifications of the momentum equations, the proposed method also yields a substantially reduced velocity error away from the interface.

These trends are further confirmed in Fig. 9, where the errors in the normal-derivative jump of the velocity across the interface and in the velocity field over the entire domain are shown relative to their analytical solutions. While the forcing error diverges for the original method and exhibits slow convergence for the hybrid and proposed methods, convergence of the forcing is not required for convergence of the velocity. The velocity error plots indicate that both the original and hybrid methods are first-order accurate, with and without including grid points near the interface. In contrast, the proposed method achieves an order of accuracy between 1.5 and 2 away from the interface for the larger grid spacings considered, as indicated by the error plots excluding near-interface points and by the 2-norm errors. We believe that this reduction in accuracy to the neglect of the second term on the right-hand side of (25), which has a significant impact on the continuity forcing term but can be neglected in the momentum equation and in the Poisson problem for the grid spacings considered here. A detailed analysis of this error source and strategies for its mitigation

are the subject of ongoing work.

4. Conclusion

The accuracy of IB methods utilizing regularized delta functions, or continuous forcing IB methods, is typically restricted to first order when the IB forcing is computed to satisfy a constraint at the IB, formulated through kernel-based interpolation with a regularized delta function kernel. Furthermore, the computation of IB forcing in the projection-based solution of these IB formulations becomes ill-conditioned when the ratio of the spacing between IB markers to the grid spacing is approximately 1 or less. This is because the system becomes a discrete analogue of a Fredholm integral equation of the first kind, which is inherently ill-posed.

This paper introduces a modification to the governing and constraint equations of IB methods with regularized delta functions, enhancing both the order of accuracy and the conditioning of the force computation in a projection-based solution approach. The modification is derived by adopting the methodology of Eldredge [3], which views the IB solution as a composition of exterior and interior solutions, each multiplied by an indicator function. By applying the product rules of the operators on these composite solutions, the governing equations for the exterior and interior solutions, multiplied again by the indicator functions, are reformulated as an equation for the composite solution plus terms for each differential operator involving the gradient of the indicator function, i.e., the Dirac Delta function distributed on the IB surface. In the discrete version, a smooth indicator function is computed such that its finite-difference gradient is the regularized delta function. Our proposed method extends this approach by recognizing that terms involving regularized delta functions do not reduce to surface-only quantities, as would be the case for a singular Dirac delta function, but instead interact with the solution over a finite width away from the interface. We model the solution variation over this region as a Taylor series about the IB points, leading to a modification of the original IB forcing term in the governing equation and the addition of new terms in both the governing and constraint equations where the solution is interpolated. This results in the system for the force computation becoming a discrete analogue of a Fredholm integral equation of the second kind, which is well-posed.

We derive and test our method for two specific cases: a Dirichlet Poisson problem and an incompressible Navier-Stokes problem with no-slip boundary conditions. Numerical experiments for the Poisson problem reveal that the order of accuracy of the spatial discretization increases from first order to second order when considering the solution outside the support of the regularized delta functions. Inside this support, the solution accuracy remains limited to first order. For the Navier-Stokes problem, a similar trend is observed, with the order of accuracy of the solution outside the delta function support ranging between 1.5 and 2. We attribute this limited accuracy improvement to an approximation error when replacing the gradient of the discrete indicator fields with a regularized delta function distribution. For both problems, the conditioning of the method significantly improves, resulting in a smoother forcing solution that requires lower computational effort to solve. The proposed method can be readily adapted to different problems and boundary conditions using the same derivation approach.

The accuracy results of our proposed method suggest that continuous forcing IB methods are not confined to first-order accuracy and indicate the potential to achieve exact second-order or higher accuracy for a wide variety of PDE problems, provided that the gradient of the discrete indicator functions can be approximated more accurately by the regularized delta functions.

5. Acknowledgments

The authors thank Prof. Jeff Eldredge for the helpful discussions. This work was supported by the United States Air Force Office of Scientific Research (AFOSR MURI FA9550-23-1-0299).

Appendix A. Discrete operations and tools

In this section, we formulate the discrete operations, identities, and shorthands used in this work. We will make use of a discrete scalar field s, vector field $\mathbf{v}$, and tensor field $\mathbf{T}$, where

$$\mathsf{s} \in \mathbb{R}^{\mathcal{C}}, \qquad \mathbf{v} = \begin{pmatrix} \mathsf{v}_x \\ \mathsf{v}_y \\ \mathsf{v}_z \end{pmatrix} \in \mathbb{R}^{\mathcal{F}}, \qquad \mathbf{T} = \begin{pmatrix} \mathsf{T}_{11} & \mathsf{T}_{12} & \mathsf{T}_{13} \\ \mathsf{T}_{21} & \mathsf{T}_{22} & \mathsf{T}_{23} \\ \mathsf{T}_{31} & \mathsf{T}_{32} & \mathsf{T}_{33} \end{pmatrix} \in \mathbb{R}^{\mathcal{D}}, \tag{A.1}$$

and their composite counterparts $\overline{\mathsf{s}} = \mathsf{H}_C^+ \circ \mathsf{s}^+ + \mathsf{H}_C^- \circ \mathsf{s}^-$, $\overline{\mathbf{v}} = \mathbf{H}_{\mathcal{F}}^+ \circ \mathbf{v}^+ + \mathbf{H}_{\mathcal{F}}^- \circ \mathbf{v}^-$, and $\overline{\mathbf{T}} = \mathbf{H}_{\mathcal{D}}^+ \circ \mathbf{T}^+ + \mathbf{H}_{\mathcal{D}}^- \circ \mathbf{T}^-$.

We will also use discrete scalar data s, vector data $\mathbf{v}$, and second-order tensor data $\mathbf{T}$ on the IB points, where

$$\mathrm{s} \in \mathbb{R}^{\mathcal{S}}, \qquad \mathbf{v} = \begin{pmatrix} \mathrm{v}_x \\ \mathrm{v}_y \\ \mathrm{v}_z \end{pmatrix} \in \mathbb{R}^{\mathcal{V}}, \qquad \mathbf{T} = \begin{pmatrix} \mathrm{T}_{11} & \mathrm{T}_{12} & \mathrm{T}_{13} \\ \mathrm{T}_{21} & \mathrm{T}_{22} & \mathrm{T}_{23} \\ \mathrm{T}_{31} & \mathrm{T}_{32} & \mathrm{T}_{33} \end{pmatrix} \in \mathbb{R}^{\mathcal{T}}, \tag{A.2}$$

Throughout, we will use second-order central differences and linear interpolations.

Appendix A.1. Grid data transformations

If we use the notations I_x, I_y, I_z to denote the interpolations of a field in the three Cartesian directions by taking the two-point average and placing the result midway, we can formulate the following interpolation operations:

- Interpolation and expansion of a scalar on cell centers to a vector on cell faces

$$^{\mathcal{F}}\mathsf{I}_C\mathsf{s} = \begin{pmatrix} \mathsf{I}_x\mathsf{s} \\ \mathsf{I}_y\mathsf{s} \\ \mathsf{I}_z\mathsf{s} \end{pmatrix} \tag{A.3}$$

- Interpolation and contraction of a vector on cell faces to a scalar on cell centers

$$^{C}\mathsf{I}_{\mathcal{F}}\mathbf{v} = \mathsf{I}_x\mathbf{v} + \mathsf{I}_y\mathbf{v} + \mathsf{I}_z\mathbf{v} \tag{A.4}$$

- Interpolation and expansion of a vector on cell faces to a tensor in the cell tensor space

$$\begin{aligned} ^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v} =& ^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\left(\mathbf{v}^{(1)} \circ 1_{\mathcal{F}_x} + \mathbf{v}^{(2)} \circ 1_{\mathcal{F}_y} + \mathbf{v}^{(3)} \circ 1_{\mathcal{F}_z}\right) && \text{(A.5)} \\ =& \mathsf{I}_x\mathbf{v}^{(1)} \circ 1_{\mathcal{D}_{11}} + \mathsf{I}_y\mathbf{v}^{(1)} \circ 1_{\mathcal{D}_{12}} + \mathsf{I}_z\mathbf{v}^{(1)} \circ 1_{\mathcal{D}_{13}} + \\ & \mathsf{I}_x\mathbf{v}^{(2)} \circ 1_{\mathcal{D}_{21}} + \mathsf{I}_y\mathbf{v}^{(2)} \circ 1_{\mathcal{D}_{22}} + \mathsf{I}_z\mathbf{v}^{(2)} \circ 1_{\mathcal{D}_{23}} + && \text{(A.6)} \\ & \mathsf{I}_x\mathbf{v}^{(3)} \circ 1_{\mathcal{D}_{31}} + \mathsf{I}_y\mathbf{v}^{(3)} \circ 1_{\mathcal{D}_{32}} + \mathsf{I}_z\mathbf{v}^{(3)} \circ 1_{\mathcal{D}_{33}} \end{aligned}$$

- Interpolation and contraction of a tensor in the cell tensor space to a vector on cell faces

$$\begin{aligned} ^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\mathbf{T} =& ^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\Big(\mathbf{T}^{(11)} \circ 1_{\mathcal{D}_{11}} + \mathbf{T}^{(12)} \circ 1_{\mathcal{D}_{12}} + \mathbf{T}^{(13)} \circ 1_{\mathcal{D}_{13}} + \\ & \mathbf{T}^{(21)} \circ 1_{\mathcal{D}_{21}} + \mathbf{T}^{(22)} \circ 1_{\mathcal{D}_{22}} + \mathbf{T}^{(23)} \circ 1_{\mathcal{D}_{23}} + && \text{(A.7)} \\ & \mathbf{T}^{(31)} \circ 1_{\mathcal{D}_{31}} + \mathbf{T}^{(32)} \circ 1_{\mathcal{D}_{32}} + \mathbf{T}^{(33)} \circ 1_{\mathcal{D}_{33}}\Big) \\ =& \Big(\mathsf{I}_x\mathbf{T}^{(11)} + \mathsf{I}_y\mathbf{T}^{(12)} + \mathsf{I}_z\mathbf{T}^{(13)}\Big) \circ 1_{\mathcal{F}_x} + \\ & \Big(\mathsf{I}_x\mathbf{T}^{(21)} + \mathsf{I}_y\mathbf{T}^{(22)} + \mathsf{I}_z\mathbf{T}^{(23)}\Big) \circ 1_{\mathcal{F}_y} + && \text{(A.8)} \\ & \Big(\mathsf{I}_x\mathbf{T}^{(31)} + \mathsf{I}_y\mathbf{T}^{(32)} + \mathsf{I}_z\mathbf{T}^{(33)}\Big) \circ 1_{\mathcal{F}_z} \end{aligned}$$

Appendix A.2. Interpolations of products

- Interpolation of an element-wise scalar product in the x-direction (similar for y- and z-directions):

$$\begin{aligned} \{\mathsf{I}_x(\mathsf{s}_1 \circ \mathsf{s}_2)\}_{i+\frac{1}{2},j,k} &= \frac{\{\mathsf{s}_1\}_{i+1,j,k}\{\mathsf{s}_2\}_{i+1,j,k} + \{\mathsf{s}_1\}_{i,j,k}\{\mathsf{s}_2\}_{i,j,k}}{2} \\ &= \frac{\{\mathsf{s}_1\}_{i+1,j,k} + \{\mathsf{s}_1\}_{i,j,k}}{2}\frac{\{\mathsf{s}_2\}_{i+1,j,k} + \{\mathsf{s}_2\}_{i,j,k}}{2} \\ &\quad + \frac{\{\mathsf{s}_1\}_{i+1,j,k} - \{\mathsf{s}_1\}_{i,j,k}}{2}\frac{\{\mathsf{s}_2\}_{i+1,j,k} - \{\mathsf{s}_2\}_{i,j,k}}{2} \\ &= \{\mathsf{I}_x\mathsf{s}_1 \circ \mathsf{I}_x\mathsf{s}_2\}_{i+\frac{1}{2},j,k} + \frac{\Delta x^2}{4}\{\mathsf{D}_x\mathsf{s}_1 \circ \mathsf{D}_x\mathsf{s}_2\}_{i+\frac{1}{2},j,k} \end{aligned} \tag{A.9}$$

- Interpolation and expansion of an element-wise scalar product on cell centers to a vector on cell faces:

$$ {}^{\mathcal{F}}\mathsf{I}_C\left(\mathsf{s}_1 \circ \mathsf{s}_2\right) = {}^{\mathcal{F}}\mathsf{I}_C\mathsf{s}_1 \circ {}^{\mathcal{F}}\mathsf{I}_C\mathsf{s}_2 + \frac{\Delta\mathbf{x}^2}{4} \circ \mathsf{G}\mathsf{s}_1 \circ \mathsf{G}\mathsf{s}_2 \tag{A.10} $$

- Interpolation and expansion of an element-wise vector product on cell faces to a tensor in the cell tensor space:

$$ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\left(\mathbf{v}_1 \circ \mathbf{v}_2\right) = {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}_1 \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}_2 + \frac{\Delta\mathbf{X}^2}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{v}_1 \circ \mathsf{G}_{\mathcal{F}}\mathbf{v}_2 \tag{A.11} $$

Appendix A.3. Derivatives of products

- The finite-difference derivative of the element-wise product of two scalar fields in the x-direction (similar for y- and z-directions):

$$ \begin{aligned} \left\{\mathsf{D}_x\left(\mathsf{s}_1 \circ \mathsf{s}_2\right)\right\}_{i+\frac{1}{2},j,k} &= \frac{\{\mathsf{s}_1\}_{i+1,j,k}\{\mathsf{s}_2\}_{i+1,j,k} - \{\mathsf{s}_1\}_{i,j,k}\{\mathsf{s}_2\}_{i,j,k}}{\Delta x} \\ &= \frac{\{\mathsf{s}_1\}_{i+1,j,k} + \{\mathsf{s}_1\}_{i,j,k}}{2}\frac{\{\mathsf{s}_2\}_{i+1,j,k} - \{\mathsf{s}_2\}_{i,j,k}}{\Delta x} \\ &\quad + \frac{\{\mathsf{s}_1\}_{i+1,j,k} - \{\mathsf{s}_1\}_{i,j,k}}{\Delta x}\frac{\{\mathsf{s}_2\}_{i+1,j,k} + \{\mathsf{s}_2\}_{i,j,k}}{2} \\ &= \left\{\mathsf{I}_x\mathsf{s}_1 \circ \mathsf{D}_x\mathsf{s}_2 + \mathsf{I}_x\mathsf{s}_2 \circ \mathsf{D}_x\mathsf{s}_1\right\}_{i+\frac{1}{2},j,k} \end{aligned} \tag{A.12} $$

- Using the previous relation, one can show that the gradient of the product of two scalars fields, analogous to the continuous product rule $\nabla\left(s_1 s_2\right) = s_1\nabla s_2 + s_2\nabla s_1$, is:

$$ \mathsf{G}\left(\mathsf{s}_1 \circ \mathsf{s}_2\right) = {}^{\mathcal{F}}\mathsf{I}_C\mathsf{s}_1 \circ \mathsf{G}\mathsf{s}_2 + {}^{\mathcal{F}}\mathsf{I}_C\mathsf{s}_2 \circ \mathsf{G}\mathsf{s}_1 \tag{A.13} $$

- Similarly, the gradient of the element-wise product of a scalar field (interpolated to cell faces) and vector field, analogous to the continuous product rule $\nabla\left(s\boldsymbol{v}\right) = s\nabla\boldsymbol{v} = \boldsymbol{v}\nabla s$, is:

$$ \mathsf{G}_{\mathcal{F}}\left({}^{\mathcal{F}}\mathsf{I}_C\mathsf{s} \circ \mathbf{v}\right) = {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}{}^{\mathcal{F}}\mathsf{I}_C\mathsf{s} \circ \mathsf{G}_{\mathcal{F}}\mathbf{v} + {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v} \circ \mathsf{G}_{\mathcal{F}}{}^{\mathcal{F}}\mathsf{I}_C\mathsf{s} \tag{A.14} $$

- The divergence of the element-wise product of a scalar field (interpolated to the cell faces) and a vector field, analogous to the continuous product rule $\nabla\cdot\left(s\boldsymbol{v}\right) = \boldsymbol{v}\cdot\nabla s + s\nabla\cdot\boldsymbol{v}$, is:

$$ \begin{aligned} \left\{\mathsf{D}\left({}^{\mathcal{F}}\mathsf{I}_C\mathsf{s} \circ \mathbf{v}\right)\right\}_{i,j,k} &= \frac{1}{\Delta x}\left(\frac{\{\mathsf{s}\}_{i+1,j,k} + \{\mathsf{s}\}_{i,j,k}}{2}\{\mathsf{v}_x\}_{i+\frac{1}{2},j,k} - \frac{\{\mathsf{s}\}_{i,j,k} + \{\mathsf{s}\}_{i-1,j,k}}{2}\{\mathsf{v}_x\}_{i-\frac{1}{2},j,k}\right) \\ &\quad + \frac{1}{\Delta y}\left(\frac{\{\mathsf{s}\}_{i,j+1,k} + \{\mathsf{s}\}_{i,j,k}}{2}\{\mathsf{v}_y\}_{i,j+\frac{1}{2},k} - \frac{\{\mathsf{s}\}_{i,j,k} + \{\mathsf{s}\}_{i,j-1,k}}{2}\{\mathsf{v}_y\}_{i,j-\frac{1}{2},k}\right) \\ &\quad + \frac{1}{\Delta z}\left(\frac{\{\mathsf{s}\}_{i,j,k+1} + \{\mathsf{s}\}_{i,j,k}}{2}\{\mathsf{v}_z\}_{i,j,k+\frac{1}{2}} - \frac{\{\mathsf{s}\}_{i,j,k} + \{\mathsf{s}\}_{i,j,k-1}}{2}\{\mathsf{v}_z\}_{i,j,k-\frac{1}{2}}\right) \\ &= \frac{1}{2}\left(\frac{\{\mathsf{s}\}_{i+1,j,k} - \{\mathsf{s}\}_{i,j,k}}{\Delta x}\{\mathsf{v}_x\}_{i+\frac{1}{2},j,k} + \frac{\{\mathsf{s}\}_{i,j,k} - \{\mathsf{s}\}_{i-1,j,k}}{\Delta x}\{\mathsf{v}_x\}_{i-\frac{1}{2},j,k}\right) \\ &\quad + \frac{1}{2}\left(\frac{\{\mathsf{s}\}_{i,j+1,k} - \{\mathsf{s}\}_{i,j,k}}{\Delta y}\{\mathsf{v}_y\}_{i,j+\frac{1}{2},k} + \frac{\{\mathsf{s}\}_{i,j,k} - \{\mathsf{s}\}_{i,j-1,k}}{\Delta y}\{\mathsf{v}_y\}_{i,j-\frac{1}{2},k}\right) \\ &\quad + \frac{1}{2}\left(\frac{\{\mathsf{s}\}_{i,j+1,k} - \{\mathsf{s}\}_{i,j,k}}{\Delta z}\{\mathsf{v}_z\}_{i,j,k+\frac{1}{2}} + \frac{\{\mathsf{s}\}_{i,j,k} - \{\mathsf{s}\}_{i,j-1,k}}{\Delta z}\{\mathsf{v}_z\}_{i,j,k-\frac{1}{2}}\right) \\ &\quad + \{\mathsf{s}\}_{i,j,k}\left(\frac{\{\mathsf{v}_x\}_{i+\frac{1}{2},j,k} - \{\mathsf{v}_x\}_{i-\frac{1}{2},j,k}}{\Delta x} + \frac{\{\mathsf{v}_y\}_{i,j+\frac{1}{2},k} - \{\mathsf{v}_y\}_{i,j-\frac{1}{2},k}}{\Delta y} + \frac{\{\mathsf{v}_z\}_{i,j,k+\frac{1}{2}} - \{\mathsf{v}_z\}_{i,j,k-\frac{1}{2}}}{\Delta z}\right) \\ &= \left\{{}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\left(\mathbf{v} \circ \mathsf{G}\mathsf{s}\right)\right\}_{i,j,k} + \{\mathsf{s} \circ \mathsf{D}\mathbf{v}\}_{i,j,k} \end{aligned} \tag{A.15} $$

- Similarly, one can show that the divergence of the product of a scalar and a tensor field, analogous to the continuous product rule $\nabla \cdot (s\boldsymbol{T}) = \boldsymbol{T} \cdot \nabla s + s\nabla \cdot \boldsymbol{T}$, is:

$$\mathsf{D}_{\mathcal{D}}({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}{}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}\mathsf{s} \circ \mathbf{T}) = {}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\Big(\mathbf{T} \circ \mathsf{G}_{\mathcal{F}}\left({}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}\mathsf{s}\right)\Big) + {}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}\mathsf{s} \circ \mathsf{D}_{\mathcal{D}}\mathbf{T} \tag{A.16}$$

Appendix A.4. Differential operators applied to composite fields

The Heaviside/indicator/characteristic fields on the different cell spaces are related as follows:

$$\mathbf{H}^{\pm}_{\mathcal{F}} := {}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}\mathsf{H}^{\pm}_{\mathcal{C}} \tag{A.17}$$

$$\mathbf{H}^{\pm}_{\mathcal{D}} := {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{H}^{\pm}_{\mathcal{F}} \tag{A.18}$$

It follows that $\mathbf{H}^{\pm}_{\mathcal{D}} = \left(\mathbf{H}^{\pm}_{\mathcal{D}}\right)^{\top}$.

Note that on uniform grids:

$$\mathsf{G}_{\mathcal{F}}{}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}\mathsf{s} = \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathsf{G}\mathsf{s}\right)^{\top} \tag{A.19}$$

$$\mathsf{G}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}} \neq {}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\mathsf{G}_{\mathcal{F}} \tag{A.20}$$

$${}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}\mathsf{G} = \mathsf{D}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}} \tag{A.21}$$

We can then work out the following operations:

- Gradient of a composite scalar field:

$$\mathsf{G}\overline{\mathsf{s}} = \mathbf{H}^{+}_{\mathcal{F}} \circ \mathsf{G}\mathsf{s}^{+} + \mathsf{H}^{-}_{\mathcal{F}} \circ \mathsf{G}\mathsf{s}^{-} + \mathsf{G}\mathsf{H}^{+}_{\mathcal{C}} \circ {}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}\left(\mathsf{s}^{+} - \mathsf{s}^{-}\right) \tag{A.22}$$

- Gradient of a composite vector field:

$$\mathsf{G}_{\mathcal{F}}\overline{\mathbf{v}} = \mathbf{H}^{+}_{\mathcal{D}} \circ \mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} + \mathbf{H}^{-}_{\mathcal{D}} \circ \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-} + \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\left(\mathbf{v}^{+} - \mathbf{v}^{-}\right) \tag{A.23}$$

$$= \mathbf{H}^{+}_{\mathcal{D}} \circ \mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} + \mathbf{H}^{-}_{\mathcal{D}} \circ \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-} + \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathsf{G}\mathsf{H}^{+}_{\mathcal{C}}\right)^{\top} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\left(\mathbf{v}^{+} - \mathbf{v}^{-}\right) \tag{A.24}$$

- Divergence of a composite vector field:

$$\mathsf{D}\overline{\mathbf{v}} = \mathsf{H}^{+}_{\mathcal{C}} \circ \mathsf{D}\mathbf{v}^{+} + \mathsf{H}^{-}_{\mathcal{C}} \circ \mathsf{D}\mathbf{v}^{-} + {}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\left(\mathsf{G}\mathsf{H}^{+}_{\mathcal{C}} \circ \left(\mathbf{v}^{+} - \mathbf{v}^{-}\right)\right) \tag{A.25}$$

- Divergence of a composite tensor field:

$$\mathsf{D}_{\mathcal{D}}\overline{\mathbf{T}} = \mathbf{H}^{+}_{\mathcal{F}} \circ \mathsf{D}_{\mathcal{D}}\mathbf{T}^{+} + \mathbf{H}^{-}_{\mathcal{F}} \circ \mathsf{D}_{\mathcal{D}}\mathbf{T}^{-} + {}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\Big(\mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathbf{T}^{+} - \mathbf{T}^{-}\right)\Big) \tag{A.26}$$

$$= \mathbf{H}^{+}_{\mathcal{F}} \circ \mathsf{D}_{\mathcal{D}}\mathbf{T}^{+} + \mathbf{H}^{-}_{\mathcal{F}} \circ \mathsf{D}_{\mathcal{D}}\mathbf{T}^{-} + {}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\left(\left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathsf{G}\mathsf{H}^{+}_{\mathcal{C}}\right)^{\top} \circ \left(\mathbf{T}^{+} - \mathbf{T}^{-}\right)\right) \tag{A.27}$$

- Divergence of the gradient, or the Laplacian, of a composite scalar field:

$$\mathsf{L}\overline{\mathsf{s}} = \mathsf{D}\mathsf{G}\overline{\mathsf{s}} \tag{A.28}$$

$$= \mathsf{H}^{+}_{\mathcal{C}} \circ \mathsf{L}\mathsf{s}^{+} + \mathsf{H}^{-}_{\mathcal{C}} \circ \mathsf{L}\mathsf{s}^{-} + {}^{\mathcal{C}}\mathsf{I}_{\mathcal{F}}\left(\mathsf{G}\mathsf{H}^{+}_{\mathcal{C}} \circ \left(\mathsf{G}\mathsf{s}^{+} - \mathsf{G}\mathsf{s}^{-}\right)\right) + \mathsf{D}\left(\mathsf{G}\mathsf{H}^{+}_{\mathcal{C}} \circ {}^{\mathcal{F}}\mathsf{I}_{\mathcal{C}}\left(\mathsf{s}^{+} - \mathsf{s}^{-}\right)\right) \tag{A.29}$$

- Laplacian of a composite vector field

$$\mathsf{L}_{\mathcal{F}}\overline{\mathbf{v}} = \mathsf{D}_{\mathcal{D}}\mathsf{G}_{\mathcal{F}}\overline{\mathbf{v}} \tag{A.30}$$

$$= \mathbf{H}^{+}_{\mathcal{F}} \circ \mathsf{L}_{\mathcal{F}}\mathbf{v}^{+} + \mathbf{H}^{-}_{\mathcal{F}} \circ \mathsf{L}_{\mathcal{F}}\mathbf{v}^{-} + {}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\left(\left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathsf{G}\mathsf{H}^{+}_{\mathcal{C}}\right)^{\top} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)$$

$$+ \mathsf{D}_{\mathcal{D}}\left(\left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathsf{G}\mathsf{H}^{+}_{\mathcal{C}}\right)^{\top} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\left(\mathbf{v}^{+} - \mathbf{v}^{-}\right)\right) \tag{A.31}$$

- Time derivative of a composite velocity field:

$$\frac{\mathrm{d}\overline{\mathbf{v}}}{\mathrm{d}t} = \mathbf{H}^{+}_{\mathcal{F}} \circ \frac{\mathrm{d}\mathbf{v}^{+}}{\mathrm{d}t} + \mathbf{H}^{-}_{\mathcal{F}} \circ \frac{\mathrm{d}\mathbf{v}^{-}}{\mathrm{d}t} + \frac{\mathrm{d}\mathbf{H}^{+}_{\mathcal{F}}}{\mathrm{d}t} \circ (\mathbf{v}^{+} - \mathbf{v}^{-}) \tag{A.32}$$

$$\frac{\mathrm{D}\mathbf{H}^{+}_{\mathcal{F}}}{\mathrm{D}t} = \frac{\mathrm{d}\mathbf{H}^{+}_{\mathcal{F}}}{\mathrm{d}t} + \mathbf{v} \cdot \nabla \mathbf{H}_{\mathcal{F}} \tag{A.33}$$

$$\frac{\mathrm{d}\mathbf{H}^{+}_{\mathcal{F}}}{\mathrm{d}t} = -\mathbf{v} \cdot \mathsf{G}\mathbf{H}_{\mathcal{F}} \tag{A.34}$$

- Convective derivative of the composite velocity field in divergence form using Eq. (A.11), Eq. (A.16), $(\mathbf{H}^{\pm}_{\mathcal{D}})^{\top} = \mathbf{H}^{\pm}_{\mathcal{D}}$ and $(\mathbf{H}^{\pm}_{\mathcal{D}})^{2} = \mathbf{H}^{\pm}_{\mathcal{D}}(1 - \mathbf{H}^{\pm}_{\mathcal{D}})$:

$$\begin{aligned}
&\mathsf{D}_{\mathcal{D}}\left(\overline{\mathbf{v}} \otimes \overline{\mathbf{v}}\right) \\
&= \mathsf{D}_{\mathcal{D}}\left(\left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\overline{\mathbf{v}}\right)^{\top} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\overline{\mathbf{v}}\right) \\
&= \mathsf{D}_{\mathcal{D}}\left(\mathbf{H}^{+}_{\mathcal{D}} \circ \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{+}\right)^{\top} \circ \mathbf{H}^{+}_{\mathcal{D}} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{+}\right) + \mathsf{D}_{\mathcal{D}}\left(\mathbf{H}^{+}_{\mathcal{D}} \circ \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{+}\right)^{\top} \circ \mathbf{H}^{-}_{\mathcal{D}} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{-}\right) \\
&\quad + \mathsf{D}_{\mathcal{D}}\left(\mathbf{H}^{-}_{\mathcal{D}} \circ \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{-}\right)^{\top} \circ \mathbf{H}^{+}_{\mathcal{D}} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{+}\right) + \mathsf{D}_{\mathcal{D}}\left(\mathbf{H}^{-}_{\mathcal{D}} \circ \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{-}\right)^{\top} \circ \mathbf{H}^{-}_{\mathcal{D}} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{-}\right) \\
&= \mathsf{D}_{\mathcal{D}}\left(\mathbf{H}^{+}_{\mathcal{D}} \circ \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{+}\right)^{\top} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{+} + \mathbf{H}^{-}_{\mathcal{D}} \circ \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{-}\right)^{\top} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\mathbf{v}^{-}\right) \\
&\quad - \mathsf{D}_{\mathcal{D}}\left(\mathbf{H}^{+}_{\mathcal{D}} \circ \mathbf{H}^{-}_{\mathcal{D}} \circ \left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\left(\mathbf{v}^{+} - \mathbf{v}^{-}\right)\right)^{\top} \circ {}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\left(\mathbf{v}^{+} - \mathbf{v}^{-}\right)\right) \\
&\quad + \mathsf{D}_{\mathcal{D}}\left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\overline{\mathbf{v}}^{\top} \circ \left(\frac{\Delta\mathbf{X}^{2}}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)\right) \\
&\quad + \mathsf{D}_{\mathcal{D}}\left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\overline{\mathbf{v}} \circ \left(\frac{\Delta\mathbf{X}^{2}}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)^{\top}\right) \\
&\quad - \mathsf{D}_{\mathcal{D}}\left(\left(\frac{\Delta\mathbf{X}^{2}}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)^{\top} \circ \left(\frac{\Delta\mathbf{X}^{2}}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)\right) \\
&= \mathbf{H}^{+}_{\mathcal{F}} \circ \mathsf{D}_{\mathcal{D}}\left(\mathbf{v}^{+} \otimes \mathbf{v}^{+}\right) + \mathbf{H}^{-}_{\mathcal{F}} \circ \mathsf{D}_{\mathcal{D}}\left(\mathbf{v}^{-} \otimes \mathbf{v}^{-}\right) \\
&\quad + {}^{\mathcal{F}}\mathsf{I}_{\mathcal{D}}\left(\mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\left(\mathbf{v}^{+} \otimes \mathbf{v}^{+}\right) - \left(\mathbf{v}^{-} \otimes \mathbf{v}^{-}\right)\right)\right) \\
&\quad - \mathsf{D}_{\mathcal{D}}\left(\mathbf{H}^{+}_{\mathcal{D}} \circ \mathbf{H}^{-}_{\mathcal{D}} \circ \left(\mathbf{v}^{+} - \mathbf{v}^{-}\right) \otimes \left(\mathbf{v}^{+} - \mathbf{v}^{-}\right)\right) \\
&\quad + \mathsf{D}_{\mathcal{D}}\left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\overline{\mathbf{v}}^{\top} \circ \left(\frac{\Delta\mathbf{X}^{2}}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)\right) \\
&\quad + \mathsf{D}_{\mathcal{D}}\left({}^{\mathcal{D}}\mathsf{I}_{\mathcal{F}}\overline{\mathbf{v}} \circ \left(\frac{\Delta\mathbf{X}^{2}}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)^{\top}\right) \\
&\quad - \mathsf{D}_{\mathcal{D}}\left(\left(\frac{\Delta\mathbf{X}^{2}}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)^{\top} \circ \left(\frac{\Delta\mathbf{X}^{2}}{4} \circ \mathsf{G}_{\mathcal{F}}\mathbf{H}^{+}_{\mathcal{F}} \circ \left(\mathsf{G}_{\mathcal{F}}\mathbf{v}^{+} - \mathsf{G}_{\mathcal{F}}\mathbf{v}^{-}\right)\right)\right)
\end{aligned} \tag{A.35}$$

Appendix A.5. Taylor series construction of discrete fields and their gradients

where $\mathrm{s}, \mathrm{s}^{n}, \mathrm{s}^{t}, \mathrm{s}^{b} \in \mathbb{R}^{\mathcal{S}}$ denote the discrete approximations to s and its normal and tangential derivatives on IB points, respectively.

- Taylor series construction of the scalar field s about the point $\mathbf{X}_l$

$$\{(\mathsf{s}^{\pm})_l^{\mathrm{TS}}\}_{i,j,k} \approx \{\mathsf{s}_\Gamma^{\pm}\}_l + \{\mathbf{n}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\{\mathsf{s}_\Gamma^{n\pm}\}_l + \{\mathbf{t}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\{\mathsf{s}_\Gamma^{t\pm}\}_l + \{\mathbf{b}\}_l \cdot (\{\mathbf{x}_C\}_{i,j,k} - \{\mathbf{X}\}_l)\{\mathsf{s}_\Gamma^{b\pm}\}_l, \quad \text{(A.36)}$$

- Gradient of the Taylor series construction of the scalar field s about the point $\mathbf{X}_l$

$$\{\mathsf{G}(\mathsf{s}^{\pm})_l^{\mathrm{TS}}\}_{i,j,k} \approx \begin{pmatrix} \{\mathsf{s}_\Gamma^{n\pm}\}_l\{\mathrm{n}_x\}_l + \{\mathsf{s}_\Gamma^{t\pm}\}_l\{\mathrm{t}_x\}_l + \{\mathsf{s}_\Gamma^{b\pm}\}_l\{\mathrm{b}_x\}_l \\ \{\mathsf{s}_\Gamma^{n\pm}\}_l\{\mathrm{n}_y\}_l + \{\mathsf{s}_\Gamma^{t\pm}\}_l\{\mathrm{t}_y\}_l + \{\mathsf{s}_\Gamma^{b\pm}\}_l\{\mathrm{b}_y\}_l \\ \{\mathsf{s}_\Gamma^{n\pm}\}_l\{\mathrm{n}_z\}_l + \{\mathsf{s}_\Gamma^{t\pm}\}_l\{\mathrm{t}_z\}_l + \{\mathsf{s}_\Gamma^{b\pm}\}_l\{\mathrm{b}_z\}_l \end{pmatrix}, \quad \text{(A.37)}$$

- Taylor series construction of the vector field v about the point $\mathbf{X}_l$

$$\{(\mathbf{v}^{\pm})_l^{\mathrm{TS}}\}_{i,j,k} \approx \begin{pmatrix} \{\mathrm{v}_{x,\Gamma}^{\pm}\}_l + \{\mathbf{n}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_x}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{x,\Gamma}^{n\pm}\right\}_l + \{\mathbf{t}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_x}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{x,\Gamma}^{t\pm}\right\}_l + \{\mathbf{b}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_x}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{x,\Gamma}^{b\pm}\right\}_l \\ \{\mathrm{v}_{y,\Gamma}^{\pm}\}_l + \{\mathbf{n}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_y}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{y,\Gamma}^{n\pm}\right\}_l + \{\mathbf{t}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_y}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{y,\Gamma}^{t\pm}\right\}_l + \{\mathbf{b}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_y}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{y,\Gamma}^{b\pm}\right\}_l \\ \{\mathrm{v}_{z,\Gamma}^{\pm}\}_l + \{\mathbf{n}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_z}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{z,\Gamma}^{n\pm}\right\}_l + \{\mathbf{t}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_z}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{z,\Gamma}^{t\pm}\right\}_l + \{\mathbf{b}\}_l \cdot \left(\{\mathbf{x}_{\mathcal{F}_z}\}_{i,j,k} - \{\mathbf{X}\}_l\right)\left\{\mathrm{v}_{z,\Gamma}^{b\pm}\right\}_l \end{pmatrix}, \quad \text{(A.38)}$$

- Gradient of the Taylor series construction of the vector field v about the point $\mathbf{X}_l$

$$\{\mathsf{G}_{\mathcal{F}}(\mathbf{v}^{\pm})_l^{\mathrm{TS}}\}_{i,j,k} \approx \begin{pmatrix} \left\{\mathrm{v}_{x,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_x\}_l + \left\{\mathrm{v}_{x,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_x\}_l + \left\{\mathrm{v}_{x,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_x\}_l & \left\{\mathrm{v}_{x,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_y\}_l + \left\{\mathrm{v}_{x,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_y\}_l + \left\{\mathrm{v}_{x,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_y\}_l & \left\{\mathrm{v}_{x,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_z\}_l + \left\{\mathrm{v}_{x,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_z\}_l + \left\{\mathrm{v}_{x,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_z\}_l \\ \left\{\mathrm{v}_{y,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_x\}_l + \left\{\mathrm{v}_{y,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_x\}_l + \left\{\mathrm{v}_{y,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_x\}_l & \left\{\mathrm{v}_{y,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_y\}_l + \left\{\mathrm{v}_{y,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_y\}_l + \left\{\mathrm{v}_{y,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_y\}_l & \left\{\mathrm{v}_{y,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_z\}_l + \left\{\mathrm{v}_{y,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_z\}_l + \left\{\mathrm{v}_{y,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_z\}_l \\ \left\{\mathrm{v}_{z,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_x\}_l + \left\{\mathrm{v}_{z,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_x\}_l + \left\{\mathrm{v}_{z,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_x\}_l & \left\{\mathrm{v}_{z,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_y\}_l + \left\{\mathrm{v}_{z,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_y\}_l + \left\{\mathrm{v}_{z,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_y\}_l & \left\{\mathrm{v}_{z,\Gamma}^{n\pm}\right\}_l\{\mathrm{n}_z\}_l + \left\{\mathrm{v}_{z,\Gamma}^{t\pm}\right\}_l\{\mathrm{t}_z\}_l + \left\{\mathrm{v}_{z,\Gamma}^{b\pm}\right\}_l\{\mathrm{b}_z\}_l \end{pmatrix}, \quad \text{(A.39)}$$

Appendix A.6. Regularization shorthands

- Normal-distance-weighted regularization to cell faces

$$\{\mathsf{R}_{\mathcal{F},1n}(\mathbf{v})\}_{i,j,k} \coloneqq \sum_l \begin{pmatrix} \{\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathbf{n}\}_l \cdot (\mathbf{x}_{\mathcal{F}_x} - \{\mathbf{X}\}_l)\{\mathrm{v}_x\}_l \\ \{\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathbf{n}\}_l \cdot (\mathbf{x}_{\mathcal{F}_y} - \{\mathbf{X}\}_l)\{\mathrm{v}_y\}_l \\ \{\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathbf{n}\}_l \cdot (\mathbf{x}_{\mathcal{F}_z} - \{\mathbf{X}\}_l)\{\mathrm{v}_z\}_l \end{pmatrix}\{\mathrm{S}\}_l. \quad \text{(A.40)}$$

- Regularization of a tensor using the transpose of the DDFs interpolated to the tensorspace

$$\{\mathsf{R}_{(\mathsf{I}\mathcal{F})^{\mathrm{T}}}\mathbf{T}\}_{i,j,k} \coloneqq \sum_l \begin{pmatrix} \{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{T}_{11}\}_l & \{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{T}_{12}\}_l & \{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{T}_{13}\}_l \\ \{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{T}_{21}\}_l & \{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{T}_{22}\}_l & \{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{T}_{23}\}_l \\ \{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_x,l}\}_{i,j,k}\{\mathrm{T}_{31}\}_l & \{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_y,l}\}_{i,j,k}\{\mathrm{T}_{32}\}_l & \{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_z,l}\}_{i,j,k}\{\mathrm{T}_{33}\}_l \end{pmatrix}\{\mathrm{S}\}_l, \quad \text{(A.41)}$$

- Normal-distance-weighted regularization of a tensor using the transpose of the DDFs interpolated to the tensorspace

$$\{\mathsf{R}_{(\mathsf{I}\mathcal{F})^{\mathrm{T}},1n}\mathbf{T}\}_{i,j,k} \coloneqq \sum_l \begin{pmatrix} \left\{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_x,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_C - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{11}\}_l & \left\{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_y,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_{\mathcal{E}_z} - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{12}\}_l & \left\{\mathsf{I}_x\mathsf{d}_{\mathcal{F}_z,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_{\mathcal{E}_y} - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{13}\}_l \\ \left\{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_x,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_{\mathcal{E}_z} - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{21}\}_l & \left\{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_y,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_C - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{22}\}_l & \left\{\mathsf{I}_y\mathsf{d}_{\mathcal{F}_z,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_{\mathcal{E}_x} - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{23}\}_l \\ \left\{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_x,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_{\mathcal{E}_y} - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{31}\}_l & \left\{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_y,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_{\mathcal{E}_x} - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{32}\}_l & \left\{\mathsf{I}_z\mathsf{d}_{\mathcal{F}_z,l}\right\}_{i,j,k}\{\mathbf{n}\}_l \cdot \left(\mathbf{x}_C - \{\mathbf{X}\}_l\right)\{\mathrm{T}_{33}\}_l \end{pmatrix}\{\mathrm{S}\}_l. \quad \text{(A.42)}$$

Appendix B. Solution of general saddle-point systems

A general block system (with non-singular matrix A) can be factorized as follows:

$$\begin{bmatrix} A & B_1^{\mathrm{T}} \\ B_2 & -C \end{bmatrix} = \begin{bmatrix} A & 0 \\ B_2 & S \end{bmatrix}\begin{bmatrix} I & A^{-1}B_1^{\mathrm{T}} \\ 0 & I \end{bmatrix}, \quad \text{(B.1)}$$

where

$$S \equiv -C - B_2A^{-1}B_1^{\mathrm{T}} \quad \text{(B.2)}$$

is the *Schur complement* of the matrix system and I is the identity. By this decomposition, one can develop the *Schur complement reduction* algorithm [21] for the solution of the block system

$$\begin{bmatrix} A & B_1^\top \\ B_2 & -C \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}. \tag{B.3}$$

We will refer to x as the solution vector and y as the constraint force. We define the intermediate solution vector $(x^*, y^*)^\top$ as the solution of the lower-triangular system

$$\begin{bmatrix} A & 0 \\ B_2 & S \end{bmatrix} \begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \tag{B.4}$$

and then the solution we seek can be found by back substitution of

$$\begin{bmatrix} I & A^{-1}B_1^\top \\ 0 & I \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x^* \\ y^* \end{pmatrix} \tag{B.5}$$

The algorithm we derive from this is

$$\begin{aligned} Ax^* &= r_1, \\ Sy^* &= r_2 - B_2 x^*, \\ y &= y^*, \\ x &= x^* - A^{-1}B_1^\top y. \end{aligned} \tag{B.6}$$

It is also useful to have an inverse representation of the block matrix system:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} A^{-1} + A^{-1}B_1^\top S^{-1} B_2 A^{-1} & -A^{-1}B_1^\top S^{-1} \\ -S^{-1}B_2A^{-1} & S^{-1} \end{bmatrix} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}. \tag{B.7}$$

For a more detailed discussion on the numerical solution of saddle-point systems, the reader is referred to Benzi et al. [21] and other related works cited therein.

Appendix C. Derivation of the incompressible Navier-Stokes equations for composite solutions

We start the derivation of the incompressible Navier-Stokes equations for the composite velocity field $\boldsymbol{v} \coloneqq H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-$ and pressure field $p \coloneqq H^+p^+ + H^-p^-$ by adding together the Navier-Stokes equations for the exterior and interior solutions, multiplied by their indicator fields:

$$H^+\left(\frac{\partial \boldsymbol{v}^+}{\partial t} + \nabla\cdot(\boldsymbol{v}^+\boldsymbol{v}^+)\right) + H^-\left(\frac{\partial \boldsymbol{v}^-}{\partial t} + \nabla\cdot(\boldsymbol{v}^-\boldsymbol{v}^-)\right) = H^+\left(-\nabla p^+ + \frac{1}{\mathrm{Re}}\nabla^2\boldsymbol{v}^+\right) + H^-\left(-\nabla p^- + \frac{1}{\mathrm{Re}}\nabla^2\boldsymbol{v}^-\right) \tag{C.1}$$

$$H^+\left(\nabla\cdot\boldsymbol{v}^+\right) + H^-\left(\nabla\cdot\boldsymbol{v}^-\right) = 0. \tag{C.2}$$

Then we apply the product rules for the composite velocity, noting that $\nabla H^- = -\nabla H^+$, $H^\pm H^\pm = H^\pm$, and $H^\pm H^\mp = 0$:

$$\nabla(H^+p^+ + H^-p^-) = H^+\nabla p^+ + H^-\nabla p^- + (p^+ - p^-)\nabla H^+, \tag{C.3}$$

$$\nabla(H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-) = H^+\nabla\boldsymbol{v}^+ + H^-\nabla\boldsymbol{v}^- + (\boldsymbol{v}^+ - \boldsymbol{v}^-)\nabla H^+, \tag{C.4}$$

$$\begin{aligned} \nabla^2(H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-) &= \nabla\cdot\nabla(H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-) \\ &= H^+\nabla^2\boldsymbol{v}^+ + H^-\nabla^2\boldsymbol{v}^- + (\nabla\boldsymbol{v}^+ - \nabla\boldsymbol{v}^-)\cdot\nabla H^+ + \nabla\cdot\left(\nabla H^+(\boldsymbol{v}^+ - \boldsymbol{v}^-)\right), \end{aligned} \tag{C.5}$$

$$\begin{aligned} \nabla\cdot\left((H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-)(H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-)\right) &= \nabla\cdot(H^+H^+\boldsymbol{v}^+\boldsymbol{v}^+ + H^+H^-\boldsymbol{v}^+\boldsymbol{v}^- + H^-H^+\boldsymbol{v}^-\boldsymbol{v}^+ + H^-H^+\boldsymbol{v}^-\boldsymbol{v}^-) \\ &= \nabla\cdot(H^+\boldsymbol{v}^+\boldsymbol{v}^+) + \nabla\cdot(H^-\boldsymbol{v}^-\boldsymbol{v}^-) \\ &= H^+\nabla\cdot(\boldsymbol{v}^+\boldsymbol{v}^+) + H^-\nabla\cdot(\boldsymbol{v}^-\boldsymbol{v}^-) + \nabla H^-\cdot(\boldsymbol{v}^+\boldsymbol{v}^+ - \boldsymbol{v}^-\boldsymbol{v}^-) \end{aligned} \tag{C.6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-) = H^+\frac{\mathrm{d}\boldsymbol{v}^+}{\mathrm{d}t} + H^-\frac{\mathrm{d}\boldsymbol{v}^-}{\mathrm{d}t} + (\boldsymbol{v}^+ - \boldsymbol{v}^-)\frac{\mathrm{d}H^+}{\mathrm{d}t} \tag{C.7}$$

$$\nabla\cdot(H^+\boldsymbol{v}^+ + H^-\boldsymbol{v}^-) = H^+\nabla\cdot\boldsymbol{v}^+ + H^-\nabla\cdot\boldsymbol{v}^- + (\boldsymbol{v}^+ - \boldsymbol{v}^-)\nabla H^+, \tag{C.8}$$

and substitute the expression for the time derivative of the indicator field [27, 3],

$$\frac{\mathrm{d}H^+}{\mathrm{d}t} = \nabla H^+ \cdot \dot{\boldsymbol{X}} \tag{C.9}$$

to obtain

$$\frac{\partial \boldsymbol{v}}{\partial t} + \nabla \cdot (\boldsymbol{v}\boldsymbol{v}) - \boldsymbol{F}_{\mathrm{d}\boldsymbol{v}/\mathrm{d}t} - \boldsymbol{F}_{\nabla\cdot(\boldsymbol{v}\boldsymbol{v})} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2 \boldsymbol{v} + \boldsymbol{F}_{\nabla p} - \frac{1}{\mathrm{Re}}\boldsymbol{F}_{\nabla^2 \boldsymbol{v}}, \quad \boldsymbol{x} \in \Omega, \tag{C.10}$$

$$\nabla \cdot \boldsymbol{v} - F_{\nabla\cdot(\boldsymbol{v})} = 0, \tag{C.11}$$

where

$$\begin{aligned}
\boldsymbol{F}_{\nabla p} &= \nabla H^+ (p^+ - p^-), \\
\boldsymbol{F}_{\nabla^2 \boldsymbol{v}} &= \nabla H^+ \cdot (\nabla \boldsymbol{v}^+ - \nabla \boldsymbol{v}^-) + \nabla \cdot (\nabla H^+ (\boldsymbol{v}^+ - \boldsymbol{v}^-)), \\
\boldsymbol{F}_{\nabla\cdot(\boldsymbol{v}\boldsymbol{v})} &= \nabla H^+ \cdot (\boldsymbol{v}^+\boldsymbol{v}^+ - \boldsymbol{v}^-\boldsymbol{v}^-), \\
\boldsymbol{F}_{\mathrm{d}\boldsymbol{v}/\mathrm{d}t} &= (\nabla H^+ \cdot \dot{\boldsymbol{X}})(\boldsymbol{v}^+ - \boldsymbol{v}^-), \\
F_{\nabla\cdot\boldsymbol{v}} &= \nabla H^+ \cdot (\boldsymbol{v}^+ - \boldsymbol{v}^-).
\end{aligned}$$

References

[1] Charles S Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, October 1972. ISSN 00219991. doi: 10.1016/0021-9991(72)90065-4. URL https://linkinghub.elsevier.com/retrieve/pii/0021999172900654.

[2] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A Front-Tracking Method for the Computations of Multiphase Flow. *Journal of Computational Physics*, 169(2): 708–759, May 2001. ISSN 00219991. doi: 10.1006/jcph.2001.6726. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999101967269.

[3] Jeff D. Eldredge. A method of immersed layers on Cartesian grids, with application to incompressible flows. *Journal of Computational Physics*, 448:110716, January 2022. ISSN 0021-9991. doi: 10.1016/j.jcp.2021.110716. URL https://www.sciencedirect.com/science/article/pii/S0021999121006112.

[4] Rajat Mittal and Gianluca Iaccarino. Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37 (1):239–261, January 2005. ISSN 0066-4189, 1545-4479. doi: 10.1146/annurev.fluid.37.061903.175743. URL https://www.annualreviews.org/doi/10.1146/annurev.fluid.37.061903.175743.

[5] R. P. Beyer and R. J. LeVeque. Analysis of a One-Dimensional Model for the Immersed Boundary Method. *SIAM Journal on Numerical Analysis*, 29(2):332–364, April 1992. ISSN 0036-1429. doi: 10.1137/0729022. URL https://epubs.siam.org/doi/10.1137/0729022. Publisher: Society for Industrial and Applied Mathematics.

[6] Anna-Karin Tornberg and Björn Engquist. Numerical approximations of singular source terms in differential equations. *Journal of Computational Physics*, 200(2):462–488, November 2004. ISSN 00219991. doi: 10.1016/j.jcp.2004.04.011. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999104001767.

[7] Boyce E. Griffith and Charles S. Peskin. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. *Journal of Computational Physics*, 208(1):75–105, September 2005. ISSN 00219991. doi: 10.1016/j.jcp.2005.02.011. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999105000835.

[8] David B. Stein, Robert D. Guy, and Becca Thomases. Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains. *Journal of Computational Physics*, 335:155–178, April 2017. ISSN 00219991. doi: 10.1016/j.jcp.2017.01.010. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999117300207.

[9] Rajat Mittal and Jung Hee Seo. Origin and evolution of immersed boundary methods in computational fluid dynamics. *Physical Review Fluids*, 8(10):100501, October 2023. ISSN 2469-990X. doi: 10.1103/PhysRevFluids.8.100501. URL https://link.aps.org/doi/10.1103/PhysRevFluids.8.100501.

[10] Roberto Verzicco. Immersed Boundary Methods: Historical Perspective and Future Outlook. *Annual Review of Fluid Mechanics*, 55(1):129–155, January 2023. ISSN 0066-4189, 1545-4479. doi: 10.1146/annurev-fluid-120720-022129. URL https://www.annualreviews.org/doi/10.1146/annurev-fluid-120720-022129.

[11] *A High-Fidelity Simulation Framework for Turbulent Flows with Complex (Metamaterial) Structures*, 2026. AIAA SCITECH 2026 Forum. doi: 10.2514/6.2026-2622. URL https://arc.aiaa.org/doi/abs/10.2514/6.2026-2622.

[12] Kunihiko Taira and Tim Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, August 2007. ISSN 0021-9991. doi: 10.1016/j.jcp.2007.03.005. URL https://www.sciencedirect.com/science/article/pii/S0021999107001234.

[13] Bakytzhan Kallemov, Amneet Bhalla, Boyce Griffith, and Aleksandar Donev. An immersed boundary method for rigid bodies. *Communications in Applied Mathematics and Computational Science*, 11(1):79–141, February 2016. ISSN 2157-5452, 1559-3940. doi: 10.2140/camcos.2016.11.79. URL http://msp.org/camcos/2016/11-1/p04.xhtml.

[14] Andres Goza, Sebastian Liska, Benjamin Morley, and Tim Colonius. Accurate computation of surface stresses and forces with immersed boundary methods. *Journal of Computational Physics*, 321:860–873, September 2016. ISSN 0021-9991. doi: 10.1016/j.jcp.2016.06.014. URL https://www.sciencedirect.com/science/article/pii/S002199911630239X.

[15] Hang Yu and Carlos Pantano. A regularized projection immersed boundary method for smooth boundary forces. *Journal of Computational Physics*, 496:112571, January 2024. ISSN 00219991. doi: 10.1016/j.jcp.2023.112571. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999123006666.

[16] I. Kataoka. Local instant formulation of two-phase flow. *International Journal of Multiphase Flow*, 12(5):745–758, September 1986. ISSN 03019322. doi: 10.1016/0301-9322(86)90049-2. URL https://linkinghub.elsevier.com/retrieve/pii/0301932286900492.

[17] Grétar Tryggvason, Ruben Scardovelli, and Stéphane Zaleski. *Direct Numerical Simulations of Gas–Liquid Multiphase Flows*. Cambridge University Press, Cambridge, 2011. ISBN 978-0-521-78240-1. doi: 10.1017/CBO9780511975264. URL https://www.cambridge.org/core/books/direct-numerical-simulations-of-gasliquid-multiphase-flows/C6282C4E426F95C1AC8642DA7569CF0C.

[18] Salih Ozen Unverdi and Grétar Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25–37, May 1992. ISSN 0021-9991. doi: 10.1016/0021-9991(92)90307-K. URL https://www.sciencedirect.com/science/article/pii/002199919290307K.

[19] Sebastian Liska and Tim Colonius. A fast lattice Green's function method for solving viscous incompressible flows on unbounded domains. *Journal of Computational Physics*, 316:360–384, July 2016. ISSN 00219991. doi: 10.1016/j.jcp.2016.04.023. URL https://linkinghub.elsevier.com/retrieve/pii/S002199911630064X.

[20] Andreas Wiegmann. *The explicit jump immersed interface method and interface problems for differential equations*. PhD thesis, University of Washington, 1998. URL https://www.proquest.com/dissertations-theses/explicit-jump-immersed-interface-method-problems/docview/304461214/se-2?accountid=14512.

[21] Michele Benzi, Gene H. Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, May 2005. ISSN 0962-4929, 1474-0508. doi: 10.1017/S0962492904000212. URL https://www.cambridge.org/core/product/identifier/S0962492904000212/type/journal_article.

[22] Tim Colonius and Kunihiko Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28): 2131–2146, April 2008. ISSN 00457825. doi: 10.1016/j.cma.2007.08.014. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782507003362.

[23] Diederik Beckers and Jeff D. Eldredge. Planar potential flow on Cartesian grids. *Journal of Fluid Mechanics*, 941:A19, June 2022. ISSN 0022-1120, 1469-7645. doi: 10.1017/jfm.2022.238. URL https://www.cambridge.org/core/product/identifier/S0022112022002385/type/journal_article.

[24] Zhilin Li. A Fast Iterative Algorithm for Elliptic Interface Problems. *SIAM Journal on Numerical Analysis*, 35 (1):230–254, February 1998. ISSN 0036-1429, 1095-7170. doi: 10.1137/S0036142995291329. URL http://epubs.siam.org/doi/10.1137/S0036142995291329.

[25] Zhilin Li and Ming-Chih Lai. The Immersed Interface Method for the Navier–Stokes Equations with Singular Forces. *Journal of Computational Physics*, 171(2):822–842, August 2001. ISSN 00219991. doi: 10.1006/jcph.2001.6813. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999101968135.

[26] Xiaolei Yang, Xing Zhang, Zhilin Li, and Guo-Wei He. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics*, 228(20):7821–7836, November 2009. ISSN 00219991. doi: 10.1016/j.jcp.2009.07.023. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999109004136.

[27] Damir Juric and Grétar Tryggvason. Computations of boiling flows. *International Journal of Multiphase Flow*, 24(3):387–410, April 1998. ISSN 03019322. doi: 10.1016/S0301-9322(97)00050-5. URL https://linkinghub.elsevier.com/retrieve/pii/S0301932297000505.