AIAA SciTech Forum
23-27 January 2023, National Harbor, MD & Online
AIAA SCITECH 2023 Forum

10.2514/6.2023-2480

A High Order 3D Immersed Interface Finite Difference Method for the Advection-Diffusion Equation

James Gabbard* and Wim M. van Rees†
Department of Mechanical Engineering
Massachusetts Institute of Technology, Cambridge, MA, 02139

We present a finite-difference based immersed interface method for the high-order discretization of 3D advection-diffusion problems on regular Cartesian grids. Our approach efficiently handles convex and non-convex geometries using a weighted least squares polynomial reconstruction algorithm. We analyze the stability of the approach for 2D and 3D parabolic and hyperbolic problems and demonstrate stable convergence results at third-order for advection and at fourth and sixth order for diffusion problems. Our immersed interface approach naturally handles one-sided Dirichlet or Neumann boundary conditions as well as two-sided jump boundary conditions within the same framework, opening the door to high-order treatment of 3D interface-coupled multiphysics problems. We demonstrate the capability of our approach using a 3D conjugate heat transfer problem resolved with third-order accuracy on a multi-resolution adaptive grid.

I. Introduction

Many fluid flow problems are strongly affected by interactions with a complex interface, such as a phase boundary in a multi-phase flow or the fluid-solid boundary in an aeroelastic system. Often the relevant quantities in these applications are interface quantities, particularly in an aerodynamics setting where the shear and pressure distributions on an immersed surface directly dictate the elastic response of an aerodynamic structure. Simulating these applications requires a numerical method that can efficiently and accurately impose boundary conditions on arbitrarily shaped interfaces. Body-fitted grids typically offer excellent accuracy for interface quantities, but are time-consuming to generate for complex geometries and do not generalize well to dynamically deforming boundaries. Immersed methods, on the other hand, are designed to avoid these issues. These methods locally adapt the numerical scheme to account for the presence of a boundary so that a structured mesh (typically Cartesian) that does not conform to the interface can be used. This ensures that complex moving or deforming geometries are naturally incorporated, but typically comes at the cost of reduced spatial accuracy for quantities on or near the interface.

Immersed methods generally fall into two categories: continuous forcing methods, which account for the interface by adding fictitious forcing terms to the governing PDE; and discrete forcing methods, which leave the PDE unaltered but modify the underlying discretization [1]. Continuous forcing methods offer simplicity of implementation, but typically require a mollification of the interface or temporal splitting of stiff forcing terms that limits their results to first order accuracy. Discrete forcing methods are often more complex, but by altering the interface discretization these schemes are able to achieve higher-order accuracy in both space and time, with many established discrete forcing methods achieving second-order spatial accuracy for elliptic [2, 3], hyperbolic [4], and parabolic [5] problems. Moreover, discrete forcing methods are readily generalized to more complex boundary conditions, including those involving the coupling of physical quantities on both sides of the interface.

Higher than second order convergence may be required to efficiently achieve high-fidelity simulation results, particularly for 3D problems involving sensitivity to numerical errors such as aeroacoustics, high Reynolds number flows, and long-time simulation results [6]. Consequently, there have been several notable efforts to increase the spatial accuracy achievable with a discrete forcing method. Within the context of finite volume schemes, the embedded boundary method developed in [5] provides a stable fourth order discretization of the 2D advection diffusion equation without relying on cell merging or ad-hoc strategies to treat small cells. For finite difference methods, the summation-by-parts scheme presented in [7] achieves up to fourth order accurate advection and eighth order accurate diffusion for 2D interfaces that bound a convex region, and the compact finite difference schemes of [8] and [9] achieve fourth order accuracy for the 3D Poisson equation and for the 2D Navier-Stokes equations in vorticity-velocity form. Finally, for

*PhD Candidate, jgabbard@mit.edu
†Assistant Professor, wvanrees@mit.edu

hyperbolic problems the third and fifth order simplified inverse Lax Wendroff (SILW) methods developed in [4, 10] have been successfully applied to the 2D Euler equations with moving boundaries [10] and the compressible Navier-Stokes equations with stationary boundaries [4].

The work we propose here focuses on the development of a high-order immersed interface method (IIM), a discrete forcing method designed for finite difference discretizations [11]. The choice of finite differences is aimed at reducing the complexity inherent in a high-order discrete forcing method: the use of pointwise data instead of cell averages allows for simpler implementation of both nonlinear terms and free-space discretizations with accuracy beyond second order. The use of point quantities also allows for high-order discretizations that require less geometric information on the interface, and we will demonstrate that the point-wise enforcement of boundary conditions leads to straightforward coupling schemes for interface-coupled multiphysics problems.

Within this immersed interface framework, we focus primarily on addressing the challenges posed by high-order discretizations of advection-diffusion problems around arbitrary, non-convex 3D geometries. For non-convex geometries, existing high-order IIMs based on stability optimization of 1D stencils [7, 12] are not directly applicable, as the full length of these stencils frequently does not fit within the problem domain when the interface bounds a non-convex region. This is a severe limitation, as any system with physics on two sides of a non-planar interface violates the convexity criterion on at least one side. Instead, we consider discretization schemes that rely on a weighted least-squares polynomial reconstruction which permits non-convex interfaces. We rely on a combination of rigorous stability analysis for 1D problems and exhaustive empirical tests to demonstrate stability in two and three dimensions. Our proposed discretization achieves third order convergence for pure advection problems under reasonable CFL constraints, and fourth and higher order convergence for diffusion problems. It can efficiently handle convex and non-convex geometries and avoids the complexity of using SILW techniques and compact finite-difference stencils. We further demonstrate that our immersed-interface approach naturally handles one-sided boundary conditions, such as standard Dirichlet or Neumann, as well as two-sided boundary conditions, such as jump conditions in solution values or derivatives common to multiphysics problems.

We detail our immersed interface approach in section II, focusing on the treatment of convex and non-convex geometries followed by specific boundary treatment for parabolic and hyperbolic partial differential equations. In section III we present stability and convergence results for the diffusion equation, advection equation, and advection-diffusion equation in both 2D and 3D, then draw conclusions in section IV.

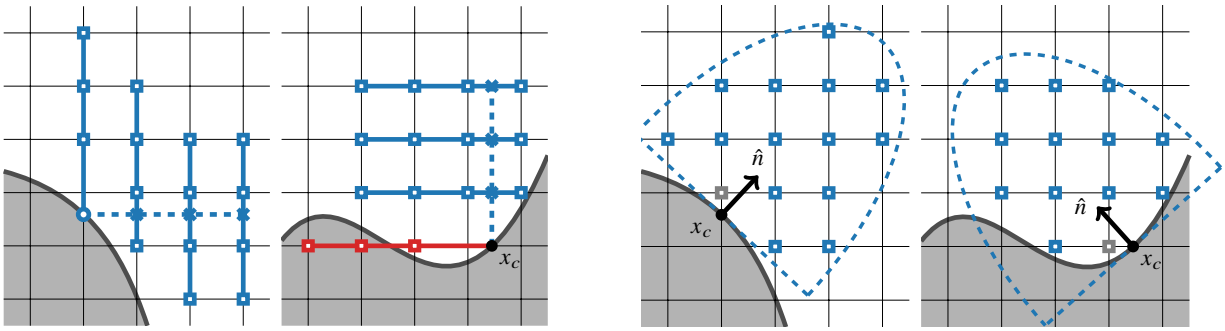
II. Methods

The work presented here extends the second order immersed interface methods developed in [2, 13, 14] to nonconvex 3D domains and higher orders of accuracy. These IIMs use free-space finite difference schemes that are composed of standard explicit 1D finite difference stencils. In general these stencils retain their full order of accuracy unless they cross a discontinuity in the the underlying PDE solution, such as a domain boundary or an immersed interface. Whenever such an intersection occurs, special treatment is required to incorporate information from the interface and retain the accuracy of the stencil. This section focuses on that treatment, first explaining the core ideas and then discussing our specific approaches for parabolic and hyperbolic PDEs in 1D as well as their extensions to 2D and 3D.

A. High-Order Polynomial Interpolants

Because the immersed interface method focuses on intersections between finite difference stencils and an interface, the minimal geometric information needed to define an interface consists of all intersections between the gridlines and the interface, collectively known as the control points, and the surface normal vector at each control point. This information can be obtained from either an explicit parametrization of the surface or from a sufficiently smooth level set function. The results presented here rely on the latter strategy, and an efficient method for determining both the intersections and normal vectors from a level set with high-order accuracy is presented in Appendix A. Once the control points have been identified, we construct a set of polynomials that approximate the PDE solution in the neighborhood of each control point. For derivatives or extrapolations taken along the intersecting gridline, a simple one-dimensional interpolant is sufficient [2, 13, 14]. For derivatives along other axes, including the normal derivative, this strategy can be extended by constructing a series of 1D interpolants in alternating coordinate directions [2] as shown in Fig. 1. Though this strategy retains the low computational cost of purely 1D interpolants, it leads to complex policies for choosing the series of interpolants and can fail when complex non-convex geometries are considered, especially in 3D.

To achieve high-order accurate interpolants that are robust to concavity in the interface, in this work we rely on multidimensional polynomial interpolants constructed using a weighted least squares approach. In d dimensions, let



(a) One dimensional interpolants. **(b) Multidimensional interpolants.**

Fig. 1 (a) The dimension-split interpolation strategy proposed in [2] constructs inexpensive 1D interpolants along two coordinate axes to calculate surface quantities, but can fail for non-convex geometries. (b) The half-elliptical multidimensional interpolant developed in this work is more robust to concave or under-resolved features.

$\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$ be a set of n interpolation points from a Cartesian grid with grid spacing h, and let $\{\phi_i\}_{i=1}^n$ be the corresponding set of samples from a smooth function $\phi(x)$. For a given set of positive weights $\{w_i\}_{i=1}^n$, the least squares interpolant $p(x)$ of degree k is defined by the minimization

$$p(x) = \underset{q(x)\in\mathcal{P}_{d,k}}{\arg\min} \sum_{x_i\in\mathcal{X}} w_i [q(x_i) - \phi_i]^2, \tag{1}$$

where $\mathcal{P}_{d,k}$ is the set of polynomials in d variables with degree less than or equal to k. So long as any basis for $\mathcal{P}_{d,k}$ is linearly independent over the interpolation points $\mathcal{X}$, this interpolant is unique and satisfies

$$\partial^\alpha p(x_e) = \partial^\alpha \phi(x_e) + O\left(h^{k+1-|\alpha|}\right) \tag{2}$$

for any evaluation point x_e and multi-index $\boldsymbol{\alpha}$. Further, the interpolant $p(x)$ is linearly related to the input data ϕ_i, so there exists a set of stencil coefficients $\{s_i\}_{i=1}^n$ such that $\partial^\alpha p(x_e) = \sum_{i=1}^n s_i \phi_i$. The linear algebra necessary to compute these coefficients is discussed in the Appendix B.

While the least squares minimization is well-defined and easy to analyze, the selection of interpolation points x_i and weights w_i leaves considerable room for experimentation. Here we postpone the discussion of weighting to Sections II.C and II.D, and focus on the nontrivial task of choosing a set of interpolation points so that any basis for $\mathcal{P}_{d,k}$ is linearly independent over $\mathcal{X}$. This requires at minimum $\dim \mathcal{P}_{d,k} = \binom{k+d}{d}$ points, and there exist useful constructions for selecting such a minimal set on a regular grid [15]. However, these constructions are complex and may fail if the available grid points are constrained by the shape of the interface. A more robust solution is to define a policy that can successfully select an adequate (but not minimal) set of points for a variety of interface geometries. In this work we select all available points that fall within a half ellipsoid centered on the control point x_c with a major radius r_n along the interface normal n_c and a minor radius r_t along any tangential direction, as shown in Fig. 1b. More precisely, let $\mathcal{G}$ be the set of all gridpoints on a regular Cartesian grid, and let $\psi(x)$ be a smooth level set function defining the immersed interface. For each control point let $\xi_c(x) = x - x_c$, and define the distance function

$$d_c(x) = \sqrt{\xi_c(x)^T \Sigma \xi_c(x)} \quad \text{with} \quad \Sigma = \frac{1}{r_n^2} n_c n_c^T + \frac{1}{r_t^2}(I - n_c n_c^T), \tag{3}$$

which ranges from $d_c(x) = 0$ at the control point to $d_c(x) = 1$ on the boundary of the ellipsoid. Using this notation we can associate two sets of interpolation points to each control point, corresponding to the semi-ellipsoids on either side of the interface:

$$\begin{aligned} \mathcal{X}_c^- &= \{x \in \mathcal{G} \text{ s.t. } \psi(x) \le 0,\ \xi_c(x)\cdot n_c \le 0, \text{ and } d_c(x) \le 1\}, \text{ and} \\ \mathcal{X}_c^+ &= \{x \in \mathcal{G} \text{ s.t. } \psi(x) > 0,\ \xi_c(x)\cdot n_c \ge 0, \text{ and } d_c(x) \le 1\}. \end{aligned} \tag{4}$$

The radii r_n and r_t are selected so that the resulting least-squares fit is well-posed for a variety of interface geometries. Suggested r_n and r_t for a range of polynomial degrees are listed in the Appendix D; they are chosen so that the

interpolation will succeed whenever the interface is smooth and the principal curvatures κ_1, κ_2 of the interface satisfy $|\kappa_i h| < 1/4$, indicating a radius of curvature greater than four gridpoints for any nonconvex feature. For a degree $k-1$ interpolant the choices generally correspond to $r_n \approx kh$ and $r_t \approx kh/2$, so that the resulting interpolation stencils resemble a one-sided stencil of order k in the normal direction and a centered stencil of order k in the tangential direction.

The sets $\mathcal{X}_c^{\pm}$ contain all grid points associated with the ellipsoids on either side of the interface. However, when enforcing boundary conditions in a PDE discretization we often wish to exclude on each side of the interface the grid point that lies closest to the control point x_c (shown in gray in Fig. 1b). This prevents any grid point in the set of interpolation points from lying arbitrarily close to x_c. Though small distances between interpolation points does not necessarily lead to an ill-conditioned least-squares system, omitting the closest point can have a notable effect on the stability of the resulting method when boundary conditions on x_c are prescribed. To shorten the notation in the sections below, we therefore define the sets $\bar{\mathcal{X}}_c^-$ and $\bar{\mathcal{X}}_c^+$, where the overbar indicates that they differ from the sets defined as in Eq. 4 by excluding the closest grid point on either side of the interface.

B. Enforcing boundary conditions

To apply this interpolation strategy to IIM-based PDE discretizations, consider a computational domain Ω with periodic or unbounded outer boundaries, and let Γ be a closed interface immersed in Ω. We assume that Γ can be described by a level set $\psi(x)$, which divides the domain into two regions

$$\Omega^+ = \{x \in \Omega \mid \psi(x) > 0\} \quad \text{and} \quad \Omega^- = \{x \in \Omega \mid \psi(x) \leq 0\}. \tag{5}$$

For $s \in \Gamma$ let $n(s)$ be the normal vector pointing into Ω^+ and away from Ω^-. Likewise, for any function $\phi(x)$ defined in a neighborhood of $s \in \Gamma$, let $[\phi](s) = \phi^+(s) - \phi^-(s)$ indicate the jump in ϕ across the interface at s; here superscripts indicate quantities defined on the side of the interface adjacent to Ω^+ or Ω^-. In this section we assume that Ω is discretized using a d-dimensional Cartesian grid with grid spacing h and unit vectors $\{e_i\}_{i=1}^d$ in each coordinate direction, and that there is a PDE solution $\phi(x)$ defined at grid points in Ω^+. For now, we consider that all points in Ω^- lie outside of the problem domain and have no associated solution value — the setting will be altered when considering multi-physics problems in the sections below.

We discretize partial derivative operators with finite difference schemes of the form

$$\frac{\partial \phi(x)}{\partial x_i} \approx \sum_{j=-\ell}^{r} a_{ij} \phi(x + jhe_i), \tag{6}$$

where $\{a_{ij}\}_{j=-\ell}^r$ are the 1D finite difference coefficients along the i-th dimension with ℓ points to the left of the evaluation point and r points to the right of the evaluation point. For grid points near the immersed interface, one or more of these 1D finite difference stencils will intersect Γ at a control point x_c, so that the free-space stencil becomes unusable. Assuming that a Dirichlet boundary condition for $\phi(x)$ is prescribed on the interface, a polynomial approximation $p_c(x)$ to the function $\phi(x)$ can be constructed using the boundary condition $\phi(x_c)$ and the domain values $\left\{\phi(x_i) \mid x_i \in \bar{\mathcal{X}}_c^+\right\}$. Each 1D finite difference stencil that intersects the boundary at x_c can then applied to the extended function

$$\phi_c(x) = \begin{cases} \phi(x), & x \in \Omega^+ \\ p_c(x), & x \in \Omega^- \end{cases}. \tag{7}$$

Provided that $p_c(x)$ is a sufficiently high order interpolant, the order of accuracy of the free space scheme in Eq. 6 will be retained even for stencils that cross the boundary at x_c. Figure 2a illustrates this procedure for a 1D discretization.

In practice, this correction is done by calculating stencil coefficients that map the boundary value $\phi(x_c)$ and a set of domain values $\left\{\phi(x_i) \mid x_i \in \bar{\mathcal{X}}_c^+\right\}$ to the value of the weighted least squares interpolant $p_c(x)$ at points along the gridline passing through x_c. This requires up to $\max(\ell, r)$ stencils at each control point, one for each stencil point in Eq. 6 that lies outside of Ω^+. Throughout this work, when the free space scheme has order of accuracy N and $p_c(x)$ has degree $k-1$ at each control point, we refer to the full IIM discretization as an (N, k) scheme for brevity.

C. Boundary treatment for parabolic PDEs

As a model parabolic PDE, we consider the diffusion equation for a scalar field $\phi(x,t)$ defined on Ω^+ with known Dirichlet boundary condition,

$$\partial_t \phi = \beta \nabla^2 \phi \ \text{ on } \ \Omega, \qquad \phi(s,t) = f(s,t) \ \text{ on } \ \Gamma, \tag{8}$$

supplemented with an initial condition $\phi(x,0) = \phi_0(x)$. The Laplacian operator in equation (8) is discretized with standard centered finite difference stencils, using stencil coefficients listed in Appendix C. Near the immersed interface Γ, the correction procedure for Dirichlet boundary conditions developed in section II.B is applied to all stencils that cross the interface. The weights for each weighted least squares interpolant are chosen to be uniform ($w_i = 1$). We note that non-uniform weighting schemes were found to be advantageous for embedded boundary finite volume discretizations of elliptic and parabolic equations in [5]. However, in our high-order IIM approach we have not observed improved spectral properties for discretizations of parabolic PDEs that use analogous weighting schemes.

Let us now instead consider Eq. 8 defined on the full domain $\Omega = \Omega^+ \cup \Omega^-$ with a diffusivity field $\beta(x)$ that is piecewise constant and discontinuous across the interface. In this case the boundary condition is replaced by jump conditions of the form

$$\begin{aligned} [\phi(s,t)] &= g(s,t) \ \text{ on } \ \Gamma, \\ [\beta \partial_n \phi(s,t)] &= h(s,t) \ \text{ on } \ \Gamma. \end{aligned} \tag{9}$$

These conditions arise naturally in problems involving discontinuous diffusivities and singular forcing terms: for $g(s,t) = h(s,t) = 0$, Eq. 9 enforces the continuity of the solution and the diffusive flux on the interface, ensuring that the integral of the scalar $\phi(x)$ over the full domain Ω is conserved. Setting $g(s,t) \neq 0$ or $h(s,t) \neq 0$ corresponds to the addition of a singular source on the interface.

For this PDE setting, the jump conditions in Eq. 9 are used to determine the value of $\phi(x)$ on each side of the interface Γ and each side of Γ is then treated using the procedure outlined above for Dirichlet boundaries. To do so, boundary values $\phi^-(x_c)$ and $\phi^+(x_c)$ are defined at each control point, and two polynomial interpolants $p_c^-(x)$ and $p_c^+(x)$ are constructed so that $p_c^\pm(x)$ interpolates the boundary value $\phi^\pm(x_c)$ and the domain values $\left\{\phi(x_i) \mid x_i \in \bar{\mathcal{X}}_c^\pm\right\}$. The boundary values are chosen so that the full two-sided interpolant $p_c(x)$ satisfies the jump conditions in Eq. 9 at x_c. Once this is done, the interpolants $p_c^+(x)$ and $p_c^-(x)$ can be used in the correction procedure for Dirichlet boundaries. This procedure is illustrated in Fig. 2b. In practice the values $\phi^\pm(x_c)$ are computed with the aid of a set of stencils. The first set of stencil coefficients $\{s_c^+, s_i^+\}$ maps the boundary value $\phi^+(x_c)$ and the domain values $\left\{\phi(x_i) \mid x_i \in \bar{\mathcal{X}}_c^+\right\}$ to the normal derivative $\partial_n \phi^+(x_c)$. The second set of coefficients $\{s_c^-, s_i^-\}$ is designed analogously to map data from the other side of the interface to the normal derivative $\partial_n \phi^-(x_c)$. The boundary values $\phi^\pm(x_c)$ can then be determined from the linear system

$$\phi^+(x_c) - \phi^-(x_c) = g(x_c) \quad \text{and} \quad \beta^+\left(s_c^+\phi^+(x_c) + \sum_{x_i \in \bar{\mathcal{X}}_c^+} s_i^+\phi^+(x_i)\right) - \beta^-\left(s_c^-\phi^-(x_c) + \sum_{x_i \in \bar{\mathcal{X}}_c^-} s_i^-\phi^-(x_i)\right) = h(x_c). \tag{10}$$

This system has two useful special cases. When there are no singular sources on the interface, $h(x_c) = g(x_c) = 0$, and the closed-form solution to Eq. 10 is

$$\phi^-(x_c) = \phi^+(x_c) = -\frac{\beta^+ \sum_{x_i \in \bar{\mathcal{X}}_c^+} s_i^+\phi^+(x_i) - \beta^- \sum_{x_i \in \bar{\mathcal{X}}_c^-} s_i^-\phi^-(x_i)}{\beta^+ s_c^+ - \beta^- s_c^-}. \tag{11}$$

In the limiting case where $\beta^- = 0$ and $\beta^+ = 1$, the jump conditions in Eq. 9 are equivalent to $\partial_n \phi^+(x) = h(x)$. Specializing Eq. 10 to this case and dropping the superscripts provides a treatment for domain boundaries with a Neumann boundary condition,

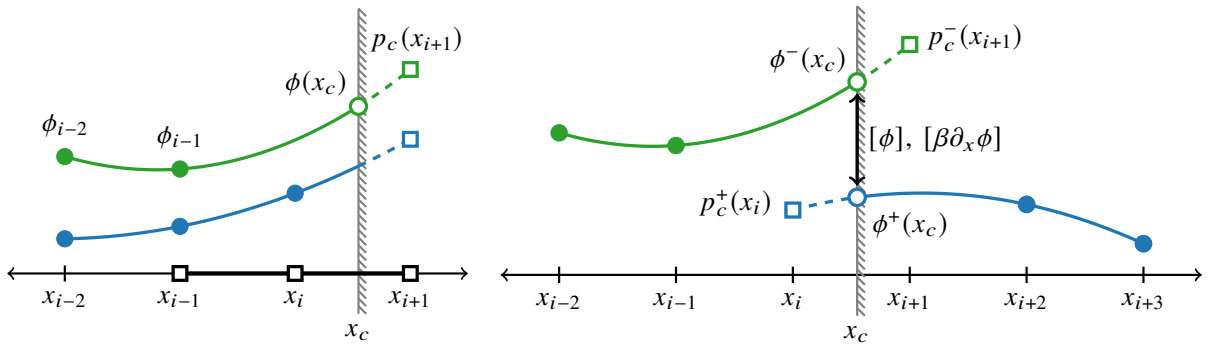
$$\phi(x_c) = \frac{1}{s_c}\left(h(x_c) - \sum_{x_i \in \bar{\mathcal{X}}_c} s_i \phi(x_i)\right). \tag{12}$$

Following the development above, this is equivalent to selecting $\phi(x_c)$ so that the resulting polynomial interpolant $p_c(x)$ exactly satisfies the Neumann boundary condition. On surfaces where a Neumann condition is prescribed, Eq. 12 is therefore used to compute the interface value $\phi(x_c)$ so that the IIM correction procedure can be performed as if $\phi(x_c)$ were a prescribed Dirichlet boundary condition.

D. Boundary treatment for hyperbolic PDEs

As a model hyperbolic PDE we consider the linear advection equation with a Dirichlet boundary condition on inflow boundaries,

$$\begin{aligned} \partial_t \phi + u(x) \cdot \nabla \phi &= 0 \ \text{ on } \ \Omega^+, \\ \phi(s,t) &= g(s,t) \ \text{ for } \ s \in \Gamma, \ u(s) \cdot n(s) > 0. \end{aligned} \tag{13}$$



(a) Extrapolation or Dirichlet conditions.

(b) Boundary treatment for jump conditions.

Fig. 2 (a) Boundary treatment for a Dirichlet boundary condition (green): a quadratic polynomial $p_c(x)$ interpolates a solution $\phi(x)$ at $\bar{\mathcal{X}}_c = \{x_{i-2}, x_{i-1}\}$ and at x_c, and the value $p(x_{i+1})$ is used when applying a three-point centered finite difference (black). When no boundary condition is prescribed (blue), $p_c(x)$ interpolates the solution at $\mathcal{X}_c = \{x_{i-2}, x_{i-1}, x_i\}$ instead. (b) Boundary treatment for an immersed interface with jump conditions. Two interpolants $p_c^+(x)$ and $p_c^-(x)$ interpolate $\phi(x)$ at $\bar{\mathcal{X}}_c^+$ and $\bar{\mathcal{X}}_c^-$, respectively, as well at the control point x_c. The boundary values $\phi^+(x_c)$ and $\phi^-(x_c)$ are chosen so that $p_c^+(x)$ and $p_c^-(x)$ satisfy the prescribed jump conditions. In multiple dimensions the interpolants pictured in (a) and (b) are replaced with weighted least-squares fits, but the principles behind the boundary treatments remain the same.

Stability is a major concern for IIM discretizations of the advection equation, and our primary focus in this section is achieving a stable discretization with at least third order accuracy. Away from the interface Eq. 13 can be discretized with an N-th order upwind finite difference scheme, which adds a slight amount of numerical dissipation and leads to a stable free-space discretization. At control points x_c that lie on the outflow boundary ($u(x_c) \cdot n(x_c) < 0$) there is no prescribed boundary condition, so the IIM correction procedure from section II.B is applied using k-th order polynomial interpolants that interpolate solution values at the points in $\mathcal{X}_c^+$. This is equivalent to applying the free-space scheme to an extrapolated field, a strategy that is known to be stable when used with upwind difference schemes [16]. Consequently, the stability of the IIM discretization schemes presented here is determined by the treatment of inflow boundary conditions on the interface.

Generally, applying our standard approach for Dirichlet boundaries to an (N, k) advection discretization results in an unstable inflow boundary treatment. This phenomena is not unique to our method: many other authors have found that special considerations are necessary to maintain stability in IIM advection discretizations. One successful strategy has been local stability optimization, which relies on boundary stencils are constructed with enough interpolation points to allow for extra degrees of freedom beyond those necessary to satisfy accuracy conditions. These additional degrees of freedom are optimized to achieve a stable discretization, typically by minimizing an objective function on each stencil that serves as heuristic indicator of stability. This approach is used in [12] to develop a stable second order immersed interface method for the 2D advection-diffusion equation, and in [7] to develop a fourth order discretization of the 2D advection equation that satisfies a summation-by-parts property. Unfortunately these optimizations are difficult to generalize beyond 1D stencils, and are not immediately applicable to interface geometries with concavity that require the use of a multidimensional interpolant.

Another successful strategy for hyperbolic boundary treatments is the simplified inverse Lax-Wendroff (SILW) boundary treatment developed in [17] and further refined in [18]. In these schemes the PDE is applied to the boundary in order to express the normal derivative of the field in terms of tangential derivatives and a prescribed boundary condition. Ghost points are then constructed near the immersed interface using a 1D polynomial extrapolation normal to the interface, and the construction of the polynomial interpolant incorporates both the boundary condition and the normal derivative. Using the stability theory of Gustafsson, Kreiss, and Sundström (GKS) [19], the authors of [20] have shown rigorously that the resulting 1D boundary treatment is conditionally stable when applied to conservative finite difference schemes of order up to thirteen, provided that the PDE and boundary condition can be differentiated a sufficiently large number of times. The analogous 2D schemes of order three and five are also shown to be stable using only one round of differentiation [18]. Unfortunately the SILW boundary treatment becomes considerably more complex in 3D and for PDEs with diffusive terms or moving interfaces, and can require extensive algebraic manipulations of the underlying

PDE.

In the following section we build off the work of Li et al. [20], using a GKS stability analysis to search for a high-order inflow boundary treatment for the 1D linear advection equation that does not rely on the SILW procedure. We further limit the search to schemes that rely on ghost points constructed with a polynomial interpolant, so that the 1D results might serve as an upper bound for what orders of accuracy are possible with multidimensional interpolants in 2D and 3D.

1. 1D Stability Analysis

To determine the stability of a given inflow boundary condition, consider a 1D linear advection equation with a Dirichlet inflow boundary

$$\begin{aligned} \partial_x \phi + \partial_t \phi = 0 \quad \text{for} \quad x \in [-\delta, \infty), \\ \phi(-\delta, t) = g(t). \end{aligned} \tag{14}$$

Here $\delta \in [0, 1]$ is an offset that describes the position of the boundary. The problem is discretized on a Cartesian mesh with unit spacing, so that with $x_j = j$ for $j \in \mathbb{Z}$. For the free space finite difference scheme, we consider standard upwind stencils with ℓ points to the left of the evaluation point and $r = \ell - 1$ points to the right of the evaluation point, which have order $2\ell - 1$ spatial accuracy; coefficients for the third and fifth order stencils are given in Appendix C. For an order k boundary treatment, a degree $k - 1$ polynomial interpolant is constructed using data at the points $\{-\delta, 1, 2, \ldots, k - 1\}$, and this interpolant is used to fill in ghost values at the points $\{-\ell, \ldots, -1\}$ before applying the free-space scheme. As in the previous section, we refer to each of these discretizations with the tuple $(2\ell - 1, k)$.

After some algebraic manipulation, the full semi-discretization can be written in the form

$$\begin{aligned} \frac{\mathrm{d}\phi_i}{\mathrm{d}t} &= \sum_{j=-\ell}^{r} a_j \phi_{i+j} \quad \text{for} \quad i \geq \ell, \\ \frac{\mathrm{d}\phi_i}{\mathrm{d}t} &= \sum_{j} b_{ij} \phi_j + c_i g(t) \quad \text{for} \quad 0 < i < \ell, \end{aligned} \tag{15}$$

where the a_j are the coefficients of the free-space scheme and the b_{ij} and c_i are boundary coefficients that depend on the interface position δ. The stability of the semi-discretization depends on the existence of eigensolution-eigenvalue pairs (ϕ, s) which satisfy

$$s\phi_i = \sum_{j=-\ell}^{r} a_j \phi_{i+j} \quad \text{for} \quad i \geq \ell, \tag{16}$$

$$s\phi_i = \sum_{j} b_{ij} \phi_j \quad \text{for} \quad 0 < i < \ell, \tag{17}$$

$$\lim_{j \to \infty} \phi_j = 0. \tag{18}$$

We limit our attention to eigenvalues s that lie outside the spectrum of the free-space scheme, which have a direct effect on the stability of the discretization. For $\mathrm{Re}\, s > 0$ the semidiscretization in Eq. 15 is unstable, and for a fully explicit method all eigenvalues s must lie within the stability region of the chosen time integrator. To calculate these eigenvalues we note that Equation 16 is a linear recurrence relation, with solutions of the form $\phi_j \propto \kappa^j$ for κ a root of the characteristic equation

$$q(\kappa, s) = s\kappa^\ell - \sum_{j=0}^{r+\ell} a_j \kappa^j = 0. \tag{19}$$

For ϕ_j to decay as $j \to \infty$, we must have $|\kappa| \leq 1$. When s is strictly outside of the free-space stability region there are no roots satisfying $|\kappa| = 1$ and ℓ roots satisfying $\kappa < 1$, so that the general solution to the linear recurrence is

$$\phi_j = \sum_{i=1}^{\ell} \sigma_i \kappa_i^j. \tag{20}$$

Substituting Eq. 20 into the boundary discretization in Eq. 17 yields an $\ell \times \ell$ linear system that determines the values of the σ_i,

$$\sum_{j=1}^{\ell} m_i(s, \kappa_j)\sigma_j = 0 \quad \text{with} \quad m_i(s, \kappa) = s\kappa^i - \sum_j b_{ij}\kappa^j. \tag{21}$$

This system has nontrivial solutions whenever the matrix M with entries $m_i(s, \kappa_j)$ has zero determinant. Thus the discretization has an eigenvalue s lying outside the stability region if there is a solution to the polynomial system of equations

$$\begin{aligned} &\det M(s, \kappa_1, \ldots, \kappa_\ell) = 0, \\ &q(\kappa_i, s) = 0 \quad \text{for} \quad i = 1, \ldots, \ell, \\ &\quad |\kappa_i| < 1 \quad \text{for} \quad i = 1, \ldots, \ell. \end{aligned} \tag{22}$$

From here the eigenvalues can be obtained numerically. We calculate the full set of solutions to the polynomial system in Eq. 22 using the software package HomotopyContinuation.jl [21], and discard any solutions that do not satisfy $|\kappa_i| < 1$. As δ is varied within the interval $[0, 1]$, the resulting eigenvalues trace out curves in the complex plane which can be plotted alongside the spectrum of the free-space difference scheme (see Fig. 3 for examples). For an IIM discretization to be stable for all interface positions δ, both the free-space spectrum and these additional eigenvalues must be contained within the stability region of the chosen time integrator.

We note that this procedure is essentially a normal mode analysis for the difference scheme, as developed in the stability theory of Godunov and Ryabenkii; Trefethen [22] provides an excellent overview of the distinction between this approach and a full GKS stability analysis. The latter is considerably richer, and treats the additional complexities that arise when there are eigenvalues satisfying $\mathrm{Re}\, s = 0$. For our purposes, the $\mathrm{Re}\, s = 0$ case rarely decides if an IIM discretizations is stable for all δ: if there is an δ for which $\mathrm{Re}\, s = 0$, then there is a nearby δ with $\mathrm{Re}\, s > 0$, and the scheme undoubtedly unstable for some interface configuration. We have also omitted the case when $q(\kappa, s)$ has a repeated root, which is considered fully in the SILW stability study in [20]. In generating the results below we did not find any distinct eigenvalues that correspond to a solution with repeated roots, which is consistent with the remark in [22] that repeated roots are "rarely relevant in practice".

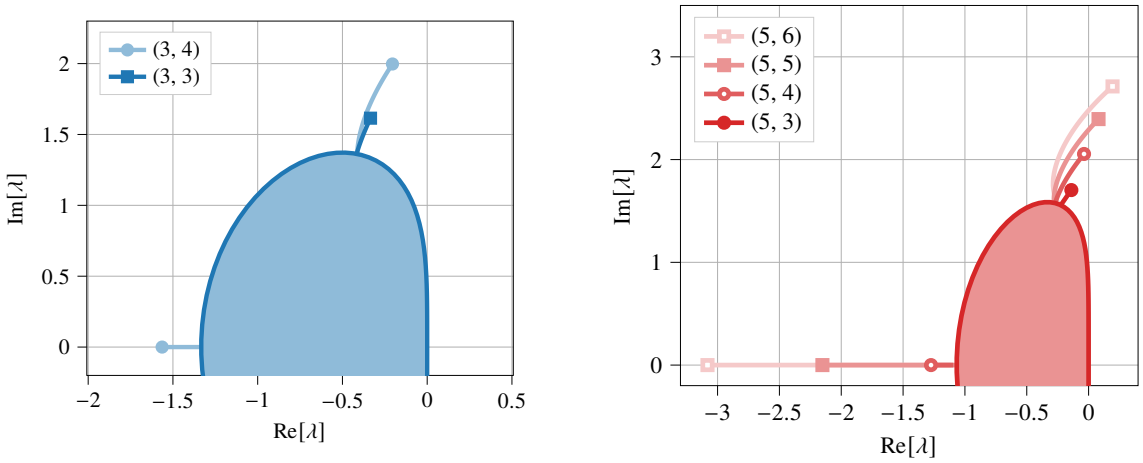
2. 1D stability results

Figure 3a plots the augmented spectrum of two different 1D IIM discretizations based on a third-order upwind stencil: a (3, 3) discretization which is second-order accurate at the boundary, and a (3, 4) discretization which is third-order accurate at the boundary. Both schemes are stable for any interface position δ. Figure 3b plots the augmented spectrum for 1D discretizations of order (5, k) for k between three and six. Of these schemes, only (5, 3) and (5, 4) are stable, giving second-order or third-order accuracy at the inflow boundary and fifth-order accuracy elsewhere. We note that these results are a departure from those in [20], which found no stable schemes of order three or higher that did not require an SILW boundary treatment. The difference can be attributed to the specific polynomial extrapolation procedures used by the authors, which differs slightly from the one presented here.

Based on the augmented spectra, Table 1 provides the maximum stable CFL number for each stable discretization when integrated with low-storage Runge-Kutta schemes of order three [23] and four [24]. For all schemes except (5, 3) with fourth order time integration there is some reduction in the stability limit compared to the free-space finite difference scheme. The effect is more severe for the third order time integrator than for the fourth order time integrator. For the (3, 4) scheme, which we focus on in later sections, the reduction is roughly 34% with third order integration and 25% with fourth order time integration.

Table 1 Maximum stable CFL for each stable IIM advection discretization when integrated with low storage Runge-Kutta methods of order three or four. The (3, F) and (5, F) entries indicate the maximum stable CFL for the free-space finite difference schemes.

Integrator	(3, F)	(3, 3)	(3, 4)	(5, F)	(5, 3)	(5, 4)
LSRK(3, 3)	1.62	1.43	1.07	1.43	1.22	0.89
LSRK(5, 4)	2.12	2.02	1.69	1.99	1.99	1.63



(a) Third-order upwind scheme.

(b) Fifth-order upwind scheme.

Fig. 3 1D stability regions for IIM discretizations based on (a) a the third order upwind stencil and (b) a fifth order upwind stencil, all using the 1D IIM boundary treatment outlined in section II.D.1. Shaded regions indicate the stability region of the free-space finite difference scheme, while the solid lines indicate additional eigenvalues beyond this region introduced by the boundary treatment.

3. Extension to 2D and 3D discretizations

Unlike in one dimension, there is no simple way to enumerate all possible configurations of a 2D or 3D interface, making a rigorous stability analysis extremely difficult. However, the 1D stability analysis provides a useful guide for what is and is not achievable in the multidimensional setting. Here we focus on adapting the (3, 4) discretization that is provably stable in 1D to advection discretizations in 2D and 3D. The free-space scheme is a third-order upwind discretization applied dimension-by-dimension, with each dimension using the 1D stencil

$$\left.\frac{\partial \phi}{\partial x}\right|_i = \begin{cases} \frac{1}{6}(\phi_{i-2} - 6\phi_{i-1} + 3\phi_i + 2\phi_{i+1}), & u_i \geq 0, \\ \frac{1}{6}(-2\phi_{i-1} - 3\phi_i + 6\phi_{i+1} + \phi_{i+2}), & u_i < 0. \end{cases} \tag{23}$$

At inflow control points satisfying $u(x_c) \cdot n(x_c) > 0$ we construct fourth order polynomial interpolants using data from the points in $\bar{\mathcal{X}}_c$ and the prescribed boundary condition at x_c. At outflow control points satisfying $u(x_c) \cdot n(x_c) \leq 0$ there is no prescribed boundary condition, so the interpolants are constructed using data from the set $\mathcal{X}_c$ only.

The procedure above allows for a third order accurate evaluation of $u \cdot \nabla\phi$, but generally leads to unstable discretizations. To ensure stability, the set of weights used in each weighted least squares interpolation (Eq. 1) must be carefully chosen. Following the work of Devendran et al. [25], we choose weights that decay rapidly with their distance from the control point, using the elliptical distance measure $d_c(x)$ defined in Eq. 3. A weight function of the form $w_i = \max(\gamma, d_c(x_i))^{-k}$ heavily de-weights interpolation points that fall outside of the region $d_c(x) < \gamma$, which is shown in [25] to improve the spectrum of a similar high-order embedded boundary finite volume scheme. Numerical experiments indicate that the (3, 4) advection scheme is stable and third-order accurate for $\gamma = 0.7$ and $k = -6$ in both two and three dimensions, and we use these parameters for the results presented below. All simulations in section III.C are performed with the LSRK(5, 4) time integrator from [24] and a CFL of 1.5, indicating that the stability limit in multiple dimensions is comparable to the 1D stability limit of $C_{\max} = 1.69$ for the (3, 4) scheme.

III. Results

A. Geometry and parameters for numerical tests

For all 2D stability and convergence results, we consider the square domain $\Omega = [0, 1]^2$ with periodic boundary conditions and an immersed interface Γ defined by a level set $\psi(x)$. In a polar coordinate system centered on the point (x_0, y_0), we choose

$$\psi(r, \theta) = -(r - r_0) - \tilde{r}\cos[5(\theta - \theta_0)], \tag{24}$$

which describes a smooth five-pointed star with center $(x_0,\ y_0)$, average radius r_0, perturbation radius $\tilde{r}$, and an angular offset of θ_0. The results presented here correspond to $x_0 = 0.502$, $y_0 = 0.503$, $r_0 = 0.36$, $\tilde{r} = 0.056$, and $\theta_0 = 0.1$, which produces an interface with curvature satisfying $|\kappa h| < 1/4$ whenever the grid spacing is $h = 1/48$ or finer. For 2D diffusion tests we consider constant diffusivities $\beta^+ = 2.2 \times 10^{-3}$ and $\beta^- = 1.1 \times 10^{-3}$ on either side of the interface, and for 2D advection convergence tests we consider both a constant velocity field $u(x) = [2, -1.5]$ and a rotating velocity field $u(x) = [-\omega(y - y_c),\ \omega(x - x_c)]$ with angular velocity $\omega = 2\pi$. The full 2D setup is illustrated in Fig. 4a.

For all 3D convergence tests we consider the cubic domain $\Omega = [0, 1]^3$ with periodic boundary conditions and a nonconvex interface consisting of a sphere that is radially perturbed by a distance proportional to a spherical harmonic:

$$\begin{aligned} \psi(\rho, \theta, \varphi) &= -(\rho - \rho_0) + \tilde{\rho} Y_5^{10}(\theta, \varphi), \\ Y_5^{10}(\theta, \varphi) &= -\frac{3}{256}\sqrt{\frac{1001}{\pi}} \cos(5\varphi) \sin^5(\theta)\Big(323\cos^5(\theta) - 170\cos^3(\theta) + 15\cos(\theta)\Big). \end{aligned} \tag{25}$$

Here (ρ, θ, φ) is a spherical coordinate system centered at $x_0 = [0.505, 0.501, 0.509]$ with azimuthal angle θ and polar angle φ. We take $\rho_0 = 0.36$ and $\tilde{\rho} = 0.045$ for all tests. Diffusion tests are performed with constant diffusivities $\beta^+ = 1.463 \times 10^{-3}$ and $\beta^- = 7.315 \times 10^{-4}$ in each region, and for advection tests we consider both a constant velocity field $u(x) = [0.62, 0.63, 0.57]$ and a rotating velocity field $u(x) = \omega \times x$ with constant angular velocity $\omega = [\pi/\sqrt{3},\ \pi/\sqrt{3},\ \pi/\sqrt{3}]$. Figure 4b illustrates the full 3D setup.

For convergence tests in $d = 2, 3$ dimensions we use the method of manufactured solutions with the sinusoidal field

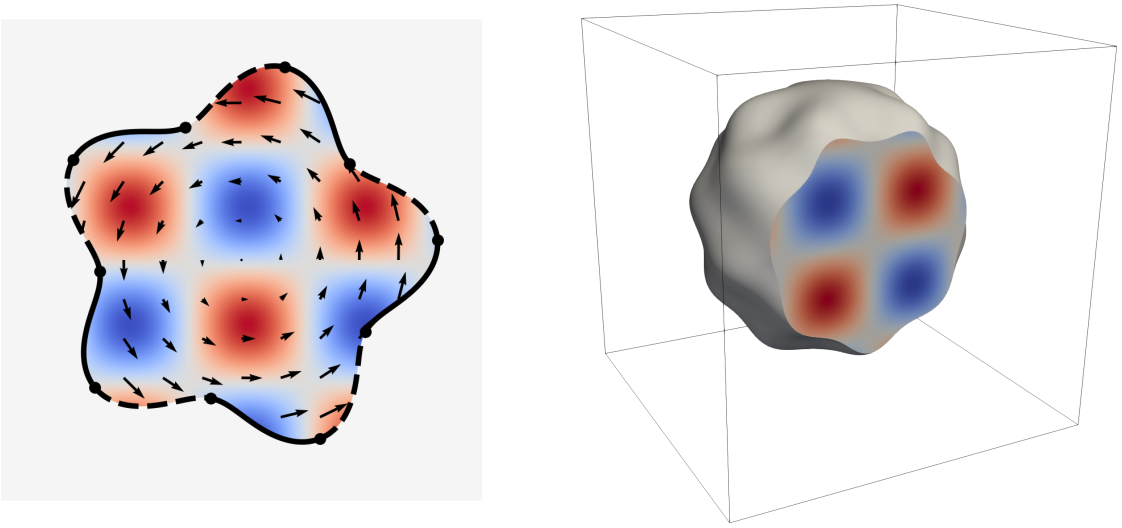
$$\phi(x, t) = \begin{cases} \exp(-\beta^+ d(4\pi)^2 t) \prod_{i=1}^d \sin(4\pi x_i), & \psi(x) > 0, \\ \exp(-\beta^- d(2\pi)^2 t) \prod_{i=1}^d \sin(2\pi x_i) + 1, & \psi(x) \le 0, \end{cases} \tag{26}$$

which satisfies the diffusion equation separately on either side of the immersed interface. For advection or advection-diffusion cases we use the same solution expressed in a coordinate system that translates or rotates with the flow. For cases with Dirichlet or Neumann boundary conditions we ignore the region with $\psi(x) < 0$ and prescribe either Eq. 26 or its normal derivative on the interface, and for cases with prescribed jump conditions we use the full domain and prescribe the jumps $[\phi]$ and $[\beta \partial_n \phi]$ on the interface. All errors are measured in the L_∞ norm over the entire problem domain, so that for a numerical solution ϕ and exact solution ϕ_e the error measure is $\epsilon_\infty = \max_{x_i \in \Omega} |\phi(x_i) - \phi_e(x_i)|$. For time-dependent cases we integrate from $t = 0$ to $t = 1$ using a five-stage fourth-order Runge-Kutta method [24]. The choice of time step varies in each test case and is discussed in the sections below.

All 3D results are generated with an implementation based on MURPHY [26], an open-source scalable software framework for PDE discretizations on 3D multiresolution grids. MURPHY provides adaptive refinement and coarsening of a block-structured grid via a wavelet-based multiresolution analysis, which is used both for generating high-order ghost points at resolution boundaries and for evaluating the adaptation criteria within each block. For convergence results we maintain uniform resolution throughout the computational domain, and only use adaptive grid refinement for the results in section III.E.

B. Stability and convergence results for the diffusion equation

Figure 5a provides convergence results for the evaluation of the Laplacian operator $\nabla^2 \phi$ using Dirichlet or Neumann boundary conditions for the $(2, 4)$, $(4, 6)$, and $(6, 8)$ IIM discretizations developed in section II.C. For this convergence test, pairing a free space scheme of order k with a polynomial interpolants of order $k + 2$ ensures that the accuracy of the free-space scheme is preserved at points near the boundary. Figure 5b provides convergence results for the time-dependent diffusion equation with Dirichlet or Neumann boundary conditions, which are calculated with a small time step of $\Delta t = 1/600$ to ensure that the spatial discretization error is dominant. For the time-dependent problem, accuracy on the interface is less crucial to achieving high-order global L_∞ norm convergence: pairs of the form (k, k) are sufficient for k-th order convergence when Dirichlet conditions are prescribed, and pairs of the form $(k, k + 1)$ are sufficient for k-th order convergence when Neumann conditions are prescribed. The gain of one or two orders of accuracy is common for immersed discretizations of the diffusion equation; see [27] for a complete discussion. Figures 5c and 5d provide the eigenvalues of the $(6, 7)$ discretization at a resolution of $h = 1/48$ for Dirichlet and Neumann boundary conditions, respectively. All eigenvalues have negative real part, and the eigenvalue with the largest negative real part differs only slightly from the expected value for the free-space discretization. This indicates that the immersed interface discretization as described in this work does not suffer from small-cell issues.



(a) 2D setup. **(b) 3D setup.**

Fig. 4 (a) Geometry, initial condition, and velocity field for 2D convergence tests with Dirichlet or Neumann boundary conditions. The rotating velocity field is shown, with solid outlines indicating an inflow boundary and dashed outlines indicating an outflow boundary. (b) Geometry used for 3D convergence tests with Dirichlet or Neumann boundaries, clipped by the plane $x_1 = 0.735$ to show the initial condition.

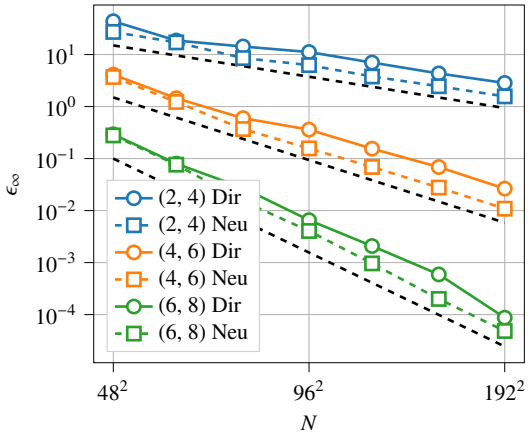
Figure 6a provides convergence results for the evaluation of the Laplacian operator with prescribed jump conditions of the form given in Eq. 9. As with the Dirichlet and Neumann boundary conditions, discretizations of the form $(k, k+2)$ converge at the order k. Figure 6b provide convergence results for the time-dependent diffusion problem, which indicate that pairs of the form $(k, k+1)$ are sufficient for k-th order convergence when jump conditions are given on the interface. Figures 6c and 6d illustrate the spectrum the discretization with $h = 1/48$ for diffusivities of $\beta^- = 1$ and $\beta^+ = 0.5$ or 10^{-4}, respectively. In both cases the eigenvalues have negative real part and their magnitude closely matches that of the free-space scheme, indicating that the sixth order IIM discretization remains stable despite a jump in the diffusivity spanning four orders of magnitude. The spectrum with the larger diffusivity ratio corresponds closely to the spectrum of the Neumann discretization shown in Fig. 5d, consistent with the observation from section II.C that our treatment of Neumann boundaries is equivalent to our treatment of jump conditions when one side of the interface has zero diffusivity.

Convergence results for the 3D diffusion equation are shown in Fig. 7a for Dirichlet or Neumann boundary conditions and in Fig. 7b for interface jump conditions. Calculating the full spectrum of each 3D discretization is infeasible due to the problem size, so we verify stability by integrating each test case with explicit time integration and a time step that is 80% as large as the maximum stable time step for the free-space finite difference scheme. For these 3D diffusion cases we use the RK4()6[2S] integrator from [28], a six-stage fourth-order Runge-Kutta method optimized for real-line stability. Consistent with the 2D results, the (k, k) discretizations with a Dirichlet boundary and the $(k, k+1)$ discretizations with a Neumann boundary condition or jump condition are stable and converge at the same rate as their free-space scheme.

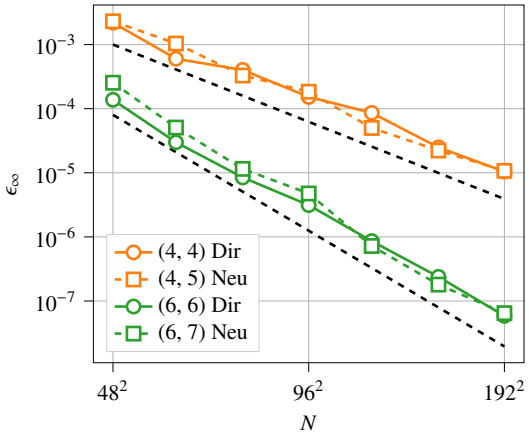
Finally, we note that for all results presented here the higher-order schemes outperform their lower-order counterparts at the same spatial resolution. Using Fig. 6b as an example, for 2D diffusion with jump conditions a sixth order scheme on a grid with 48^2 points is more accurate by an order of magnitude than a second order scheme that uses sixteen times as many grid points. This suggests that the use of higher-order immersed interface discretizations can lead to drastically reduced memory requirements in larger simulations and likely to a decrease in total computational cost as well.

C. Stability and convergence results for the advection equation

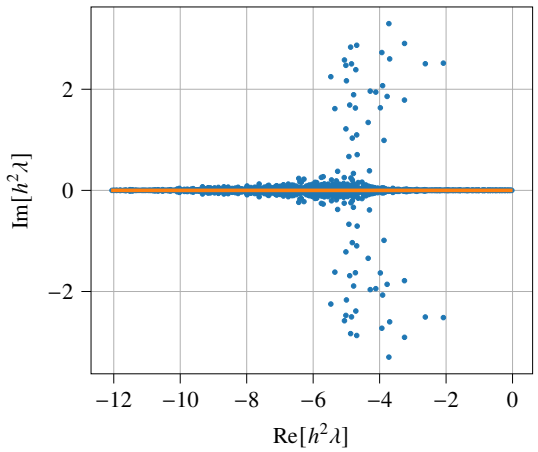
Figures 8a and 8b provides convergence results for the advection equation in 2D and 3D, respectively. The spatial discretization is the (3, 4) multidimensional advection scheme developed in section II.D.3. Time integration is performed



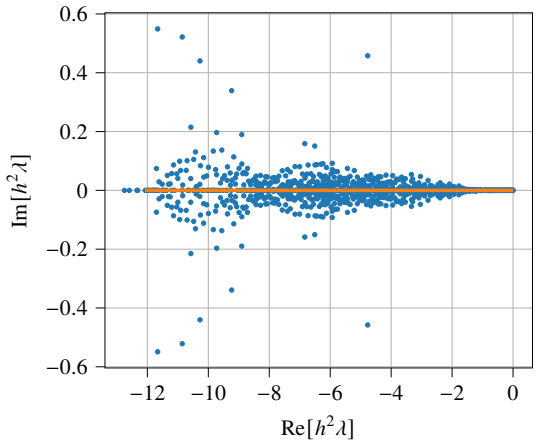
(a) Laplacian operator with Dirichlet or Neumann BCs.



(b) Scalar diffusion with Dirichlet or Neumann condition.

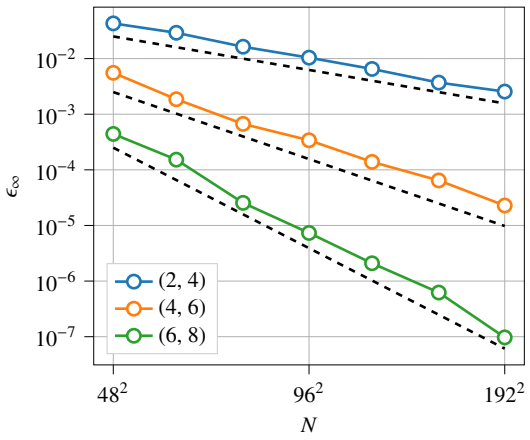


(c) Spectrum of the (6, 6) Dirichlet discretization

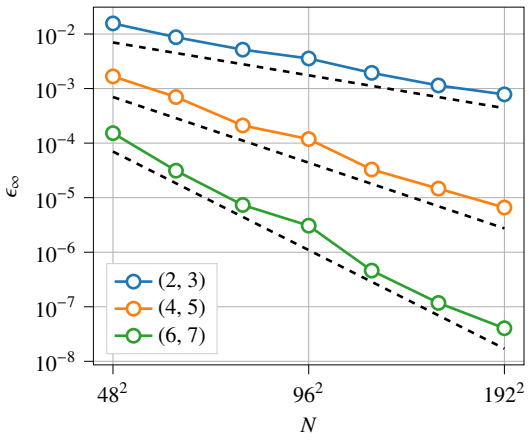


(d) Spectrum of the (6, 7) Neumann discretization

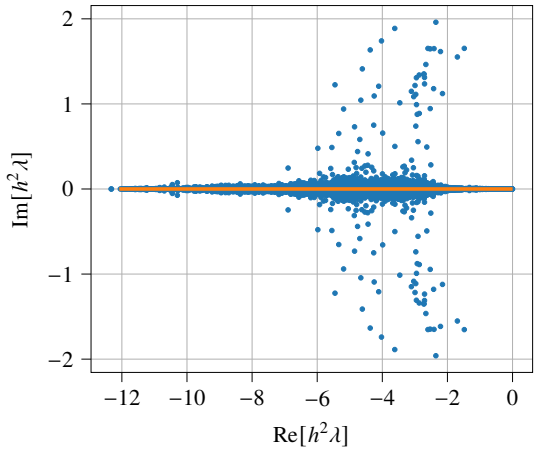
Fig. 5 (a) Convergence results for the evaluation of the Laplacian operator with Dirichlet or Neumann boundary condition. (b) Convergence results for the scalar diffusion equation with Dirichlet or Neumann boundary conditions. Dashed lines indicate the expected order of convergence for the free-space finite difference scheme. The eigenvalues of the corresponding 6th order discretizations at resolution $h = 1/48$ are shown in (c) for the Dirichlet condition and (d) for the Neumann condition, with the solid colored line indicating the spectrum of the free-space discretization.



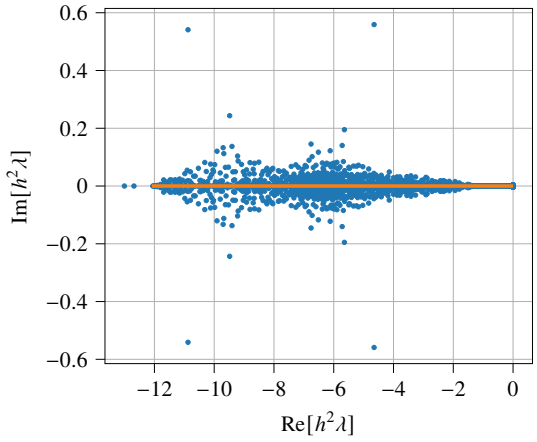
(a) Laplacian with jump conditions.



(b) Diffusion with jump conditions.

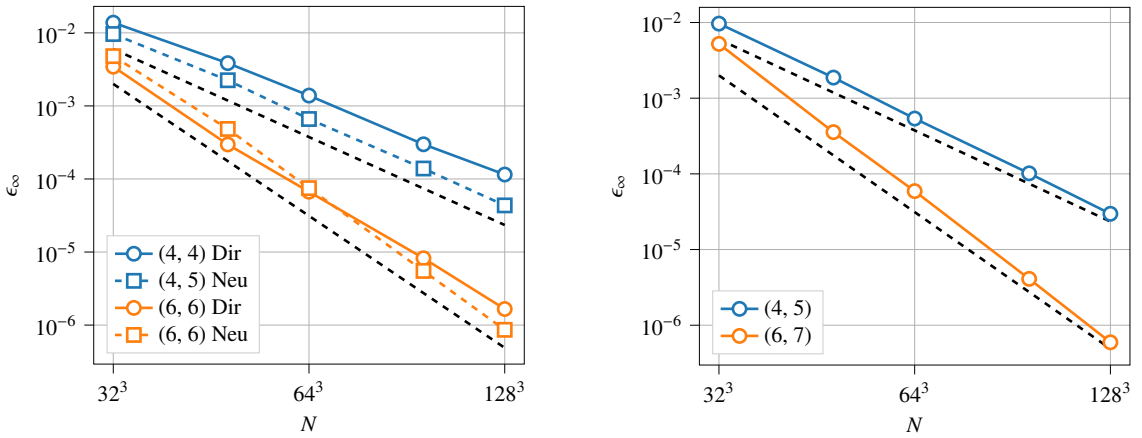


(c) Spectrum of the (6, 7) discretization, $\beta^+/\beta^- = 2$



(d) Spectrum of the (6, 7) discretization, $\beta^+/\beta^- = 10^4$

Fig. 6 Convergence results for (a) the evaluation of the Laplacian operator with interface jump conditions and (b) time-dependent diffusion with interface jump conditions. Dashed lines indicate the order of the free-space scheme. The spectrum of the (6, 7) discretization with $h = 1/48$ is shown for diffusivity ratios (c) $\beta^+/\beta^- = 2$ and (d) $\beta^+/\beta^- = 10^4$, with the spectrum of the free-space scheme superimposed as a solid line.



(a) Convergence with Dirichlet or Neumann boundaries. **(b) Convergence with prescribed jump conditions.**

Fig. 7 Convergence results for the 3D diffusion equation with (a) Dirichlet or Neumann boundary conditions and (b) prescribed jump conditions. Dashed lines indicate the order of the free-space scheme.

with a CFL number of $C = \max \|u\|_1 \Delta t/h = 1.5$, which is comparable to the 1D stability limit of $C_{\max} = 1.69$ established in section II.D.2 for this scheme and time integrator. For both constant and rotating velocity fields the discretization remains stable with no added dissipation and we observe clean third order convergence in the L_∞ norm. The spectrum of the 2D discretization with $h = 1/48$ is given in Fig. 8c for a constant velocity field and in Fig. 8d for a rotating velocity field. In both cases there are no eigenvalues with negative real part, further confirming that the 2D discretization is conditionally stable with no added dissipation.

D. Stability and convergence for the advection-diffusion equation

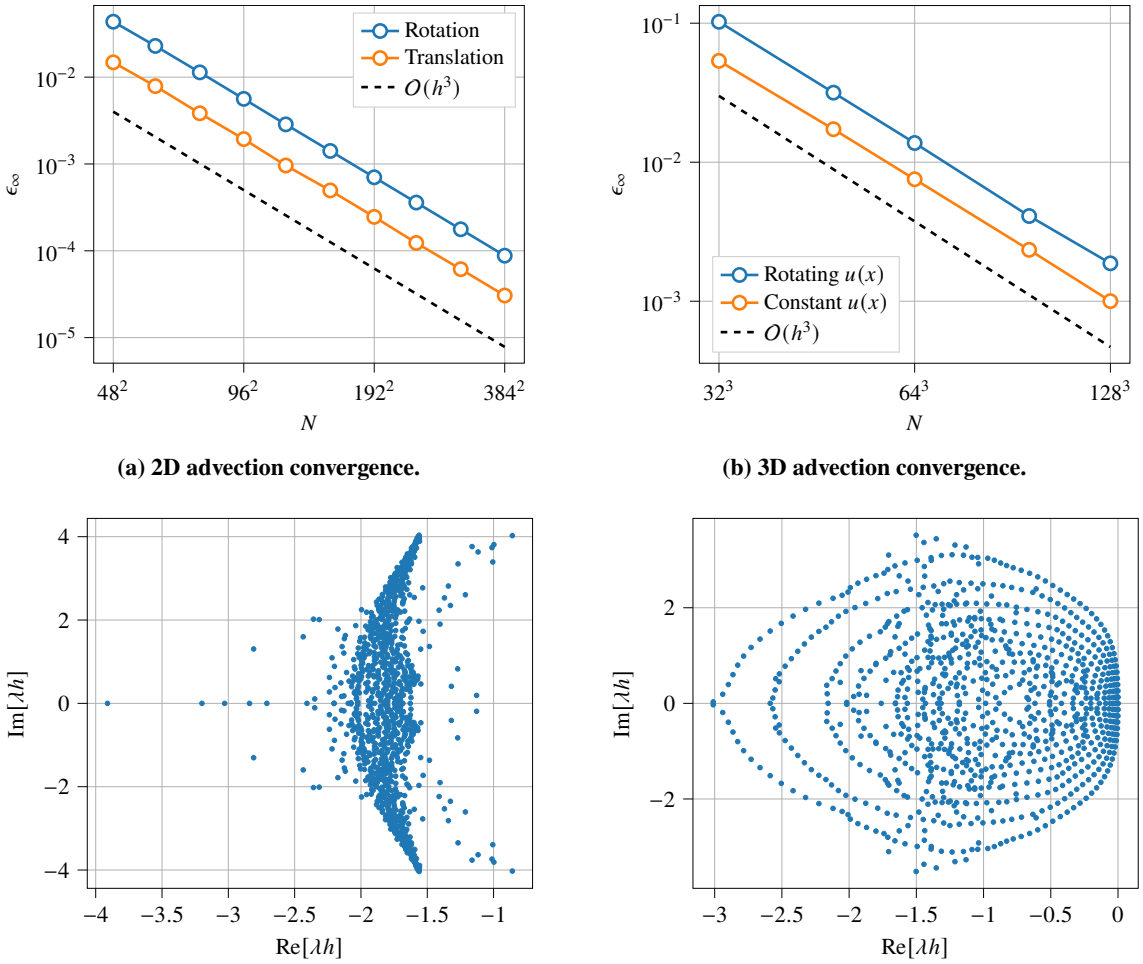
For the advection diffusion equation the relative importance of the advection term and the diffusive term is determined by the Peclet number, which we define as $\text{Pe} = \omega r_0^2/\beta^+$ in 2D and $\text{Pe} = \|\omega\|_2 \rho_0^2/\beta^+$ in 3D. We consider cases with $\text{Pe} = 50$ and $\text{Pe} = 500$ using the same geometry and rotating velocity field described in section III.A. The advection term is discretized with the (3, 4) multidimensional advection scheme that uses the prescribed Dirichlet boundary condition only at inflow boundaries, while the diffusion term is discretized with a (4, 5) scheme that uses the Dirichlet boundary condition at all points on the interface. Time integration is performed with a maximum CFL of 1.0 and maximum Fourier number of $\Delta t \beta^+/h^2 = 0.2$; in each test case the time step is chosen according to the more restrictive constraint. Figures 9a and 9b show clean third order convergence for $\text{Pe} = 50$ and $\text{Pe} = 500$ in 2D and in 3D, consistent with the third order truncation error of both the boundary and free-space schemes.

E. Conjugate heat transfer

As a final test case, we consider a model of 3D conjugate heat transfer for a solid sphere immersed in an exterior potential flow. This test case poses a number of challenges for an immersed method, which must enforce coupling conditions on the fluid-solid interface, discretize the concave interior surface of the sphere, and accurately resolve the thin thermal boundary layer that forms on the exterior surface of the sphere. For a temperature field $T(x,t)$ defined in a solid domain Ω_s and fluid domain Ω_f, we solve

$$\begin{aligned} \partial_t T &= \beta_s \nabla^2 T \quad \text{in} \quad \Omega_s, \\ \partial_t T + u \cdot \nabla T &= \beta_f \nabla^2 T \quad \text{in} \quad \Omega_f, \\ [T] = [\beta \partial_n T] &= 0 \quad \text{on} \quad \Gamma. \end{aligned} \tag{27}$$

The velocity field is divergence-free and satisfies $u \cdot n = 0$ on the interface, so that the jump conditions in Eq. 27 represent the continuity of the temperature field and the conservation of thermal energy at the interface. The solid region is a sphere with radius $r = 0.251$ centered at $x_c = [0.5, 0.5, 0.5]$, immersed in the computational domain $\Omega = [0,1] \times [0,1] \times [0,4]$. In the fluid domain $\Omega_f = \Omega \setminus \Omega_s$ we define a velocity field $u(x)$ using the potential flow solution for the flow external to a sphere, with a free stream of magnitude $U_\infty = 1$ aligned with the x_3 axis. The



(a) 2D advection convergence.

(b) 3D advection convergence.

(c) Spectrum of the 2D constant velocity discretization.

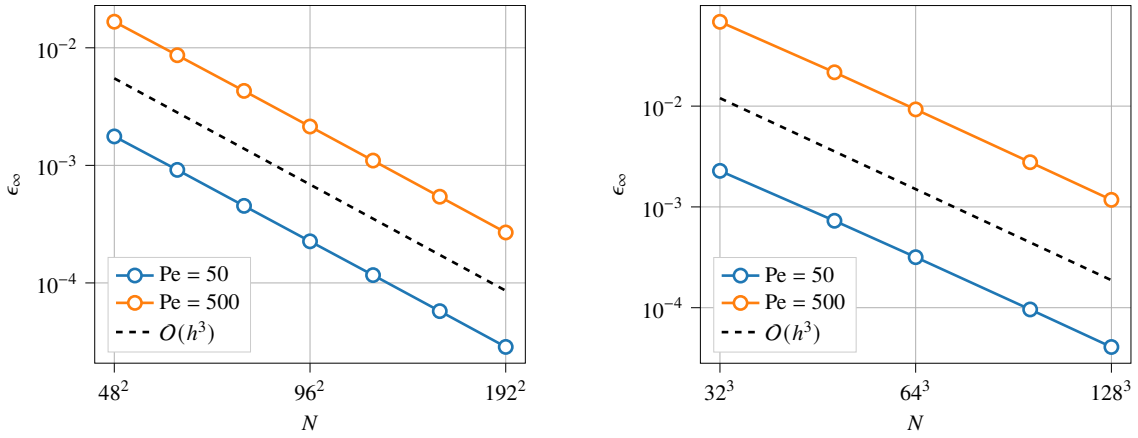
(d) Spectrum of the 2D rotating velocity discretization.

Fig. 8 Convergence results for the advection equation with constant velocity field and rotating velocity field in (a) 2D and (b) 3D. The spectrum of the 2D discretization with $h = 1/48$ is shown in (c) for the constant velocity field and in (d) for the rotating velocity field.

diffusivities in each region are $\beta_s = 0.003$ and $\beta_f = 0.001$, giving a Peclet number of $\mathrm{Pe}_D = 502$ in the fluid region and a diffusivity ratio of $\beta_s/\beta_f = 3$. At time $t = 0$ the initial temperature field is set to $T = T_0 = 1$ inside the solid region and $T = 0$ in the fluid region.

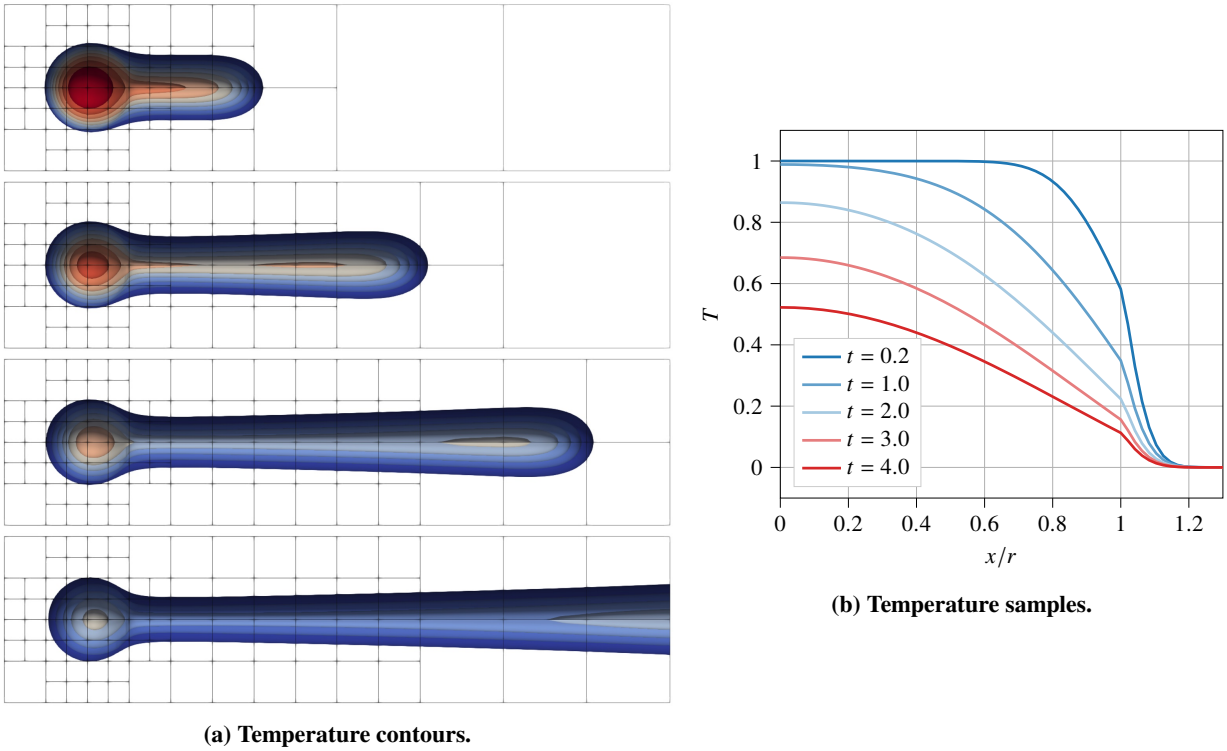
Numerically, the discretization of the advection and diffusion terms is the same as in the previous section. Zero Dirichlet boundary conditions are imposed on all exterior boundaries of Ω except for the outflow plane at $x_3 = 4$, where the field is extrapolated at sixth order in the x_3 direction before the finite difference scheme is applied. Exploiting the adaptivity in MURPHY [26], we use a sixth-order interpolating wavelet-based grid adaptation with refinement and compression tolerances of 10^{-2} and 10^{-4} in the free-space. The grid around the interface is kept at a fixed resolution of $h = 1/192$. The time step is chosen to maintain a maximum CFL number of 1.0 and a maximum Fourier number of 0.1 based on the local mesh spacing.

Figure 10a illustrates the evolution of the temperature field using evenly-spaced contours at $t = \{1.0, 2.0, 3.0, 4.0\}$. As the simulation proceeds, heat diffuses from the solid domain into the fluid domain, forming a warmer wake behind the sphere and a thin thermal boundary layer on the sphere surface. The boundary layer and the discontinuity in the temperature gradient on the fluid-solid interface are illustrated in Fig. 10b, which provides instantaneous samples of the temperature field along a line through the sphere center that is perpendicular to the flow. Our high-order IIM boundary treatment is able to sharply resolve the discontinuity, and the use of local mesh refinement provides enough spatial resolution to capture the large temperature gradients that occurs within the thin boundary layer.



(a) 2D advection-diffusion convergence. **(b) 3D advection-diffusion convergence.**

Fig. 9 Convergence results for the advection-diffusion equation with a rotating velocity field in (a) 2D and (b) 3D. Results are shown for Pe = 50 **and** Pe = 500**.**



(a) Temperature contours. **(b) Temperature samples.**

Fig. 10 (a) Temperature distribution for the conjugate heat transfer case $t = \{1.0, 2.0, 3.0, 4.0\}$ **(top to bottom). Contours represent values of** T/T_0 **between** 0.1 **and** 0.9 **in increments of** 0.1**, and are clipped by the plane** $x_1 = 0.5$**. A slice of the block-based mesh is shown in gray, with each block representing** 24^3 **grid points. (b) The temperature distribution over a line perpendicular to the free stream flow passing through the sphere center at times** $t = \{0.2, 1.0, 2.0, 3.0, 4.0\}$ **.**

IV. Conclusion

We have developed a high-order immersed interface method for the 3D advection-diffusion equation with a focus on simplicity of implementation, compatibility with multiphysics coupling, and accurate interface quantities. The method is based on point values and finite differences, allowing for simple implementation of high-order free-space schemes

and nonlinear terms. Boundary conditions or jump conditions can be enforced pointwise on one or both sides of a nonconvex interface, and numerical results indicate that the treatment of jump conditions is robust to large jumps in material properties. Convergence results demonstrate up to sixth order spatial accuracy for diffusion terms, and for advection terms the method achieves third order accuracy while allowing stable explicit time integration at CFL numbers only slightly smaller than those that determine stability for the free-space scheme.

While we have focused on the advection-diffusion equation in this work, our primary interest is a high-order immersed interface method for the incompressible Navier-Stokes equations with moving immersed bodies. This requires both an efficient 3D elliptic solver and a high-order treatment of moving boundaries. The former has been achieved with compact finite differences and preconditioned BiCGStab [9], and with high-order finite volumes and an algebraic multigrid solver [29]; these successes indicate that a similar solver should be feasible with our current approach based on explicit finite differences. High-order treatment of moving boundaries is an active research area, and some preliminary results are available in [30]. We hope to present a more comprehensive treatment in the near future.

Appendix

A. High order calculation of gridline intersections and surface geometry

The geometry processing method described here is designed for surfaces defined by a level set $\psi(x)$ that is expensive to evaluate. Consequently we assume that $\psi(x)$ is sampled on a regular Cartesian grid, and that values of ψ away from the gridpoints are unavailable. Assuming that the interface is well-resolved, an intersection exists wherever two adjacent grid points have level set values with opposite signs. To locate the intersection with $O\left(h^{2k}\right)$ accuracy, let x_0 and x_1 be the neighboring points, and let x_i be the i-th consecutive grid point along the gridline in question. Define $\bar{\psi}(x)$ to be the polynomial of order $2k-1$ that interpolates $\psi(x)$ at the $2k$ points $\{x_{-k+1}, \ldots, x_k\}$. Since $\bar{\psi}(x)$ has opposite signs at x_0 and x_1, it is guaranteed to have at least one root on the interval $[x_0, x_1]$, which is taken to be location of the intersection. To find this root numerically, the interpolant is constructed using a Barycentric Lagrange interpolation scheme [31], which is numerically stable and allows for repeated evaluations of the interpolant and its derivative with $O(k)$ operations. The root is located with a hybrid bracketing-Newton method similar to the `rtsafe` algorithm described in [32], which typically provides quadratic convergence and guarantees that the resulting root lies in the bracketing interval.

To calculate the normal vector at a given control point, $\psi(x)$ is approximated by a multidimensional polynomial interpolant. Without loss of generality, assume the intersection occurs along a gridline parallel to the x axis between points (x_0, y_0, z_0) and (x_1, y_0, z_0). The set of interpolation points chosen for an $O\left(h^{2k-1}\right)$ approximation of the normal vector is $\mathcal{X} = \{x_{-k+1}, \ldots, x_k\} \times \{y_{-k}, \ldots, y_k\} \times \{z_{-k}, \ldots, z_k\}$. Let $\tilde{\psi}(x)$ be the multidimensional interpolant of order $2k-1$ that approximates $\psi(x)$ on $\mathcal{X}$ in the unweighted least squares sense defined in section II.A; the normal vector is taken to be $n = \nabla\tilde{\psi}/\left|\nabla\tilde{\psi}\right|$. Because the shape of the interpolation region $\mathcal{X}$ is identical for each control point, the same matrix factorization can be reused when repeatedly computing these least-squares interpolants.

If information on the surface curvature is also required, it can be computed with $O\left(h^{2k-2}\right)$ accuracy using the same multidimensional interpolant. Define the surface projection matrix $P = I - nn^T$, and let H be the Hessian of the interpolant $\tilde{\psi}$ on the interface with entries $H_{ij} = \partial_{x_i}\partial_{x_j}\tilde{\psi}$. Taking the eigendecomposition of the projected Hessian $\tilde{H} = P^T HP/|\nabla\psi|$ gives three eigenvectors and their associated eigenvalues: the normal vector with associated eigenvalue 0, and the principal curvature vectors t_1 and t_2 with associated eigenvalues equal to the principal curvatures κ_1 and κ_2. If only the mean curvature is desired (as is the case for fluid interfaces with surface tension), the eigendecomposition can be dropped by noting that $\kappa_1 + \kappa_2 = \mathrm{tr}\left(\tilde{H}\right)$.

B. Calculation of stencil coefficients via least squares fitting

Section II.A outlines a weighted least squares framework for generating high-order interpolation stencils based on point values; the linear algebra required to calculate these stencil coefficients is developed here. For embedded boundary finite volume schemes, an equivalent calculation for high-order interpolants based on cell averages is presented in [25]. Let $\{x_i\}_{i=1}^n$ be a set of interpolation points, let x_e be a given evaluation point, and for any function $\phi(x)$ let $\phi = \{\phi_i\}_{i=1}^n$ be a vector containing samples taken at those points. For a stencil that uses polynomial interpolants of degree k to evaluate $\partial^\alpha\phi(x_e)$, let $m = \dim\mathcal{P}_{d,k} = \binom{k+d}{d}$, and let $\left\{b_j(x)\right\}_{j=1}^m$ be a linearly independent basis for $\mathcal{P}_{d,k}$. Any polynomial $p(x)$ in $\mathcal{P}_{d,k}$ can then be written as $p(x) = \sum_{j=1}^m c_j b_j(x)$ for some vector of coefficients c_j. Define the n by m matrix B with entries $B_{ij} = b_j(x_i)$, the n by n diagonal weight matrix W with entries $W_{ij} = w_i\delta_{ij}$, the m-vector ℓ with entries $\ell_j = \partial^\alpha b_j(x_e)$, and an n-vector s containing the unknown stencil coefficients s_i. With this

nomenclature, the least-squares interpolant $p(x)$ is determined by the vector of coefficients satisfying the least-squares system $c = \arg\min \|Bc - \phi\|_W^2$. Provided the columns of V are linearly independent, this minimization has the unique solution

$$c = \left(B^T W B\right)^{-1} B^T W \phi. \tag{28}$$

This is considerably simplified by introducing the decomposition $\sqrt{W}B = QR$ for an n by m orthogonal matrix Q and an m by m upper-triangular matrix R, which leads to the simpler expression $c = R^{-1}Q^T\sqrt{W}\phi$. The quantity of interest can then be approximated by evaluating

$$\partial^\alpha p(x_e) = \sum_{j=1}^{m} c_j \partial^\alpha q_j(x_e) = \ell^T c = \ell^T R^{-1} Q^T \sqrt{W}\phi. \tag{29}$$

The application of the stencil s to function values ϕ is given by $s^T\phi$, so Eq. 29 implies that

$$s = \sqrt{W} Q R^{-T} \ell. \tag{30}$$

Eq. 29 can be alternatively simplified by introducing a singular value decomposition, which may offer increased numerical stability in some situations. Let $\sqrt{W}B = U\Sigma V^T$, where U is an orthogonal n by m matrix, Σ is a diagonal m by m matrix, and V is an orthogonal m by m matrix. When Φ has full column rank both Σ and V are invertible, and the coefficient vector can be simplified to $c = V\Sigma^{-1}U^T\sqrt{W}f$. The resulting stencil values are given by $s = \sqrt{W}U\Sigma^{-1}V^T f$.

C. Common finite difference stencils

Table 2 provides coefficients for a centered discretization of the second derivative. Likewise, Table 3 provides 1D finite difference coefficients for an upwind discretization of the first derivative. For each stencil the eigenfunctions of the associated free-space finite difference operator are the complex exponentials e^{ikx} with associated eigenvalues $\lambda(k)$. For the standard centered schemes of even orders the eigenvalues are purely real with $-\lambda_{\max} \leq \lambda(k) \leq 0$ for a stencil-dependent constant $\lambda_{\max}$ given in the table below. For the upwind finite difference schemes, the eigenvalues satisfy $\mathrm{Re}\,\lambda(k) < 0$ for $k \neq 0$. A useful characterization of their magnitude is the largest constant C for which $Ch\lambda(k)$ is contained within the stability region of a particular time integrator; these values are listed for two low-storage Runge-Kutta schemes, a three-stage third-order method [23] and a four-stage fifth-order method [24].

Table 2 Standard finite difference stencils of even order for the second derivative.

Order	Coefficients	$h^2\lambda_{\max}$
2	$[1, -2, 1]$	4
4	$\left[-\frac{1}{12}, \frac{4}{3}, -\frac{5}{12}, \frac{4}{3}, -\frac{1}{12}\right]$	$16/3 \approx 5.33$
6	$\left[\frac{1}{90}, -\frac{3}{20}, \frac{3}{2}, -\frac{49}{18}, \frac{3}{2}, -\frac{3}{20}, \frac{1}{90}\right]$	$272/45 \approx 6.04$
8	$\left[-\frac{1}{560}, \frac{8}{315}, -\frac{1}{5}, \frac{8}{5}, -\frac{205}{72}, \frac{8}{5}, -\frac{1}{5}, \frac{8}{315}, -\frac{1}{560}\right]$	$2048/315 \approx 6.50$

Table 3 Standard upwind finite difference stencils of odd order for the first derivative.

Order	Coefficients	$C_{\max}$, LSRK(3,3)	$C_{\max}$, LSRK(5,4)
3	$\left[\frac{1}{6}, -1, \frac{1}{2}, \frac{1}{3}\right]$	1.62	2.12
5	$\left[-\frac{1}{30}, \frac{1}{4}, -1, \frac{1}{3}, \frac{1}{2}, -\frac{1}{20}\right]$	1.43	1.99

D. Radii for ellipsoidal stencil search regions

Table 4 contains the normal and tangential radii of the elliptical or ellipsoidal stencil search regions developed in section II.A. They are listed by polynomial order for both 2D and 3D geometries, and designed to guarantee an overdetermined fit on smooth surfaces satisfying the curvature constraint $|\kappa h| < 1/4$, where in 2D κ is the curvature and in 3D it is the largest absolute principal curvature.

To confirm that these radii lead to an over-determined polynomial fit, each set has been tested on randomly generated 2D or 3D nonconvex surface geometries with the maximum allowable curvature. Each geometry is characterized by an intersection point with a random offset δ that is uniformly distributed over $[0, 1]$. In 2D the domain is the interior of circle with radius $4h$, with a center chosen so that the surface normal vector at the intersection has a uniform random distribution over the unit circle. In 3D, two types of surfaces are tested: the interior of a sphere with radius $1/4h$, giving $\kappa_1 h = \kappa_2 h = 1/4$ for any intersection point, and a torus with major radius $8h$ and minor radius $4h$, giving $\kappa_1 h = 1/4$ and $\kappa_2 h = -1/4$ for an intersection point on the inner equator. Each 3D geometry is given a random surface frame at the intersection (defined by the normal vector and principal curvature directions) that is uniformly distributed over the space of 3D rotations. For a given polynomial order, a set of radii are deemed successful if they lead to an overdetermined polynomial fit in 10^5 random trials for each type of trial surface.

For the 2D diffusion results in section III.B we also consider interpolation stencils of order 8, which generally cannot be guaranteed to fit within a circle of constant curvature $1/4h$. However, for geometries where the maximum curvature $1/4h$ is not sustained across the entire surface, radii of $r_n/h = 8.5$ and $r_t/4h = 4.5$ work well. Finally, note that the cost of interpolation on any set of points is linear in the number of points, so smaller stencils are preferred for computational cost. These radii have been chosen by trial and error to be close to optimal for 3D stencils, but there is room for a more rigorous optimization.

Table 4 Elliptical search radii for 2D or 3D polynomial interpolants of order three to seven with a surface curvature constraint $|\kappa_i h| < 1/4$.

Order	r_n/h	r_t/h
3	3.4	1.7
4	4.5	2.25
5	5.5	2.75
6	6.5	3.3
7	7.6	3.95

Acknowledgments

The authors gratefully acknowledge funding from the Department of Energy Advanced Scientific Computing Research (ASCR) Program, Program Manager Dr. Steven Lee, award number DE-SC0020998.

References

[1] Mittal, R., and Iaccarino, G., "Immersed boundary methods," *Annu. Rev. Fluid Mech.*, Vol. 37, 2005, pp. 239–261.

[2] Gillis, T., Winckelmans, G., and Chatelain, P., "Fast immersed interface Poisson solver for 3D unbounded problems around arbitrary geometries," *Journal of Computational Physics*, Vol. 354, 2018, pp. 403–416.

[3] Hosseinverdi, S., and Fasel, H. F., "An efficient, high-order method for solving Poisson equation for immersed boundaries: Combination of compact difference and multiscale multigrid methods," *Journal of Computational Physics*, Vol. 374, 2018, pp. 912–940.

[4] Lu, J., Fang, J., Tan, S., Shu, C.-W., and Zhang, M., "Inverse Lax–Wendroff procedure for numerical boundary conditions of convection–diffusion equations," *Journal of Computational Physics*, Vol. 317, 2016, pp. 276–300.

[5] Overton-Katz, N., Gao, X., Guzik, S. M., Antepara, O., Graves, D., and Johansen, H., "Towards a High-Order Embedded Boundary Finite Volume Method for the Incompressible Navier-Stokes Equations with Complex Geometries," *AIAA SCITECH 2022 Forum*, 2022, p. 2202.

[6] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., et al., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845.

[7] Brady, P. T., and Livescu, D., "Foundations for high-order, conservative cut-cell methods: Stable discretizations on degenerate meshes," *Journal of Computational Physics*, Vol. 426, 2021, p. 109794.

[8] Hosseinverdi, S., and Fasel, H. F., "Very high-order accurate sharp immersed interface method: application to direct numerical simulations of incompressible flows," *23rd AIAA Computational Fluid Dynamics Conference*, 2017, p. 3624.

[9] Hosseinverdi, S., and Fasel, H. F., "A fourth-order accurate compact difference scheme for solving the three-dimensional Poisson equation with arbitrary boundaries," *AIAA Scitech 2020 Forum*, 2020, p. 0805.

[10] Tan, S., and Shu, C.-W., "Inverse Lax–Wendroff procedure for numerical boundary conditions of hyperbolic equations: survey and new developments," *Advances in applied mathematics, modeling, and computational science*, Springer, 2013, pp. 41–63.

[11] Wiegmann, A., and Bube, K. P., "The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions," *SIAM Journal on Numerical Analysis*, Vol. 37, No. 3, 2000, pp. 827–862.

[12] Brehm, C., and Fasel, H. F., "A novel concept for the design of immersed interface methods," *Journal of Computational Physics*, Vol. 242, 2013, pp. 234–267.

[13] Gillis, T., Marichal, Y., Winckelmans, G., and Chatelain, P., "A 2D immersed interface Vortex Particle-Mesh method," *Journal of Computational Physics*, Vol. 394, 2019, pp. 700–718.

[14] Gabbard, J., Gillis, T., Chatelain, P., and van Rees, W. M., "An immersed interface method for the 2D vorticity-velocity Navier-Stokes equations with multiple bodies," *Journal of Computational Physics*, 2022, p. 111339.

[15] Galimberti, G., and Pereyra, V., "Numerical differentiation and the solution of multidimensional Vandermonde systems," *Mathematics of Computation*, Vol. 24, No. 110, 1970, pp. 357–364.

[16] Goldberg, M., "On a boundary extrapolation theorem by Kreiss," *Mathematics of Computation*, Vol. 31, No. 138, 1977, pp. 469–477.

[17] Tan, S., Wang, C., Shu, C.-W., and Ning, J., "Efficient implementation of high order inverse Lax–Wendroff boundary treatment for conservation laws," *Journal of Computational Physics*, Vol. 231, No. 6, 2012, pp. 2510–2527.

[18] Lu, J., Shu, C.-W., Tan, S., and Zhang, M., "An inverse Lax-Wendroff procedure for hyperbolic conservation laws with changing wind direction on the boundary," *Journal of Computational Physics*, Vol. 426, 2021, p. 109940.

[19] Gustafsson, B., Kreiss, H.-O., and Sundström, A., "Stability theory of difference approximations for mixed initial boundary value problems. II," *Mathematics of Computation*, Vol. 26, No. 119, 1972, pp. 649–686.

[20] Li, T., Shu, C.-W., and Zhang, M., "Stability analysis of the inverse Lax–Wendroff boundary treatment for high order upwind-biased finite difference schemes," *Journal of Computational and Applied Mathematics*, Vol. 299, 2016, pp. 140–158.

[21] Breiding, P., and Timme, S., "HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia," *International Congress on Mathematical Software*, Springer, 2018, pp. 458–465.

[22] Trefethen, L. N., "Group velocity interpretation of the stability theory of Gustafsson, Kreiss, and Sundström," *Journal of Computational Physics*, Vol. 49, No. 2, 1983, pp. 199–217.

[23] Williamson, J. H., "Low-storage runge-kutta schemes," *Journal of Computational Physics*, Vol. 35, No. 1, 1980, pp. 48–56.

[24] Carpenter, M. H., and Kennedy, C. A., "Fourth-order 2N-storage Runge-Kutta schemes," *NASA Technical Memorandum 109112*, 1994.

[25] Devendran, D., Graves, D., Johansen, H., and Ligocki, T., "A fourth-order Cartesian grid embedded boundary method for Poisson's equation," *Communications in Applied Mathematics and Computational Science*, Vol. 12, No. 1, 2017, pp. 51–79.

[26] Gillis, T., and van Rees, W. M., "MURPHY—A Scalable Multiresolution Framework for Scientific Computing on 3D Block-Structured Collocated Grids," *SIAM Journal on Scientific Computing*, Vol. 44, No. 5, 2022, pp. C367–C398.

[27] Johansen, H. S., *Cartesian grid embedded boundary finite difference methods for elliptic and parabolic partial differential equations on irregular domains*, University of California, Berkeley, 1997.

[28] Ketcheson, D. I., "Runge–Kutta methods with minimum storage implementations," *Journal of Computational Physics*, Vol. 229, No. 5, 2010, pp. 1763–1773.

[29] Overton-Katz, N., Gao, X., Guzik, S., Antepara, O., Graves, D. T., and Johansen, H., "A Fourth-Order Embedded Boundary Finite Volume Method for the Unsteady Stokes Equations with Complex Geometries," *arXiv:2209.02840*, 2022.

[30] Gabbard, J., "An immersed interface method for incompressible flow with moving boundaries and high order time integration," *Master's Thesis, Massachusetts Institute of Technology*, 2020.

[31] Berrut, J.-P., and Trefethen, L. N., "Barycentric lagrange interpolation," *SIAM review*, Vol. 46, No. 3, 2004, pp. 501–517.

[32] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical recipes in C: The art of scientific computing (second edition)*, Cambridge University Press, 1992.