4-15-2017

Immersed Boundary Smooth Extension (IBSE): A High-Order Method for Solving Incompressible Flows in Arbitrary Smooth Domains

David B. Stein
University of California, Davis

Robert D. Guy
University of California, Davis

Becca Thomases
University of California, Davis, bthomases@smith.edu

Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains

David B. Stein[a,*], Robert D. Guy[a], Becca Thomases[a]

[a] *Department of Mathematics, University of California, Davis, Davis, CA 95616-5270, USA*

Abstract

The Immersed Boundary method is a simple, efficient, and robust numerical scheme for solving PDE in general domains, yet for fluid problems it only achieves first-order spatial accuracy near embedded boundaries for the velocity field and fails to converge pointwise for elements of the stress tensor. In a previous work we introduced the Immersed Boundary Smooth Extension (IBSE) method, a variation of the IB method that achieves high-order accuracy for elliptic PDE by smoothly extending the unknown solution of the PDE from a given smooth domain to a larger computational domain, enabling the use of simple Cartesian-grid discretizations. In this work, we extend the IBSE method to allow for the imposition of a divergence constraint, and demonstrate high-order convergence for the Stokes and incompressible Navier-Stokes equations: up to third-order pointwise convergence for the velocity field, and second-order pointwise convergence for all elements of the stress tensor. The method is flexible to the underlying discretization: we demonstrate solutions produced using both a Fourier spectral discretization and a standard second-order finite-difference discretization.

Keywords: Embedded boundary, Immersed Boundary, Incompressible Navier-Stokes, Fourier spectral method, Complex geometry, High-order

1. Introduction

The Immersed Boundary (IB) method was originally developed for the study of moving, deformable structures immersed in a fluid, and it has been widely applied to such problems since its introduction [1–3]. Recently, the method has been adapted to more general fluid-structure problems, including the motion of rigid bodies immersed in a fluid [4] and fluid flow through a domain with either stationary boundaries or boundaries with prescribed motion [5, 6]. In this broadened context, we use the term *Immersed Boundary method* to refer only to methods in which (*i*) the boundary is treated as a Lagrangian structure embedded in a geometrically simple domain, (*ii*) the background PDE (e.g. the Navier-Stokes equations) are solved on a Cartesian grid everywhere in that domain, and (*iii*) all communication between the Lagrangian structure and the underlying PDE is mediated by convolutions with regularized δ-functions. These methods have many desirable properties: they make use of robust and efficient Cartesian-grid methods for solving the underlying PDE, are flexible to a wide range of problems, and are simple to implement, requiring minimal geometric information and processing to describe the boundary.

The IB method belongs to the broad category of methods known as *embedded boundary* (EB) methods, including the Immersed Interface [7], Ghost Fluid [8], and Volume Penalty methods [9]. These methods share a common feature: they enable solutions to PDE on nontrivial domains to be computed using efficient and robust structured-grid discretizations; yet these methods differ largely in how boundary conditions are enforced and whether or not the solution is produced in the entirety of a simple domain. Methods which

*Corresponding author
Email address: dbstein@math.ucdavis.edu (David B. Stein)

compute the solution everywhere in a d-dimensional rectangle admit the simplest discretizations and enable the use of high-order discretizations such as Fourier spectral methods. Unfortunately, this simplicity comes coupled with a fundamental difficulty: the *analytic* solution to these problems is rarely globally smooth on the entire domain. Consider the one-dimensional Poisson problem $\Delta u = f$ on the periodic interval $\mathbb{T} = [0, 2\pi]$ with Dirichlet boundary conditions $u(a) = u(b) = 0$ for $a \neq b \in \mathbb{T}$. Even if $f \in C^\infty(\mathbb{T})$, the solution u will typically display jumps in its derivative at the values $x = a$ and $x = b$. The lack of regularity in the analytic problem leads to low-order convergence in many numerical schemes, including the Immersed Boundary method: the addition of (regularized) singular forces supported at the boundary causes the solution to be C^0, and solutions are accurate only to first-order in the grid spacing Δx [4, 5].

The advantages of EB methods are substantial enough that significant effort has been expended on improving their accuracy [10–26]. Two different approaches are generally taken. The first approach involves locally altering the discretization of the PDE in the vicinity of the boundary to accommodate the lack of smoothness in the solution. One example of this approach is the Immersed Interface method [7]. Such approaches are particularly useful for interface problems where the solution is required on both sides of the embedded boundary. When the solution is only needed on one side of the interface, a second approach may be taken in which variables are redefined outside of the domain of interest to obtain higher global regularity. Improved convergence rates are achieved as a natural consequence of the properties of the discretization scheme when applied to smooth problems. Variations of this basic idea have been used by the Fourier Continuation (FC) [23–25] and the Active Penalty (AP) [26] methods to provide high order solutions to PDE on general domains.

In [27], we introduce another approach: the *Immersed Boundary Smooth Extension* (IBSE) method for the solution of the Poisson and related problems (e.g. the heat equation and Burgers equation). The IBSE method builds off of the basic framework of the direct forcing IB method. Drawing on ideas from the AP and FC methods, the IBSE method remedies the slow convergence of the direct forcing IB method by solving an auxiliary problem to extend the *unknown* solution from the physical domain into the entire computational domain. This extension is used to define the body forcing in the non-physical portion of the domain, forcing the solution to be *globally* C^k. High-order accuracy is then achieved naturally, up to $\mathcal{O}(\Delta x^{k+1})$ or the maximum accuracy supported by the underlying discretization for a smooth problem.

In this paper, we extend the work from [27] to apply to PDE requiring the imposition of a divergence constraint, including the Stokes and Navier-Stokes equations. This is a non-trivial difficulty for a high-order embedded boundary scheme based on smooth extensions. Prior work has either been restricted to the compressible Navier-Stokes equations [23], or has dealt with the pressure in an ad-hoc manner, limiting the overall accuracy of the numerical scheme to second-order [26]. These methods either extend the forcing function or solution from a prior timestep, and are thus unable to provide solutions to the Stokes problem. For time-dependent problems, they require either explicit timestepping or the use of Alternating-Direction-Implicit (ADI) schemes to take implicit timesteps, complicating the ability to achieve high-order accuracy in time.

In contrast, we work directly with the Stokes equations, rather than employing dimensional splitting or decoupling the problem into a set of Poisson equations. The smooth extensions to the solution of the Stokes equation define forcing functions in the non-physical domain in a coupled manner, and the Lagrange multipliers supported on the boundary that define these extensions are fully coupled through the Stokes problem, eliminating any need to impose artificial boundary conditions for a pressure Poisson equation. The IBSE method smoothly extends *the unknown solution*, allowing for efficient direct solutions of the steady incompressible Stokes equation and simple high-order in time implicit-explicit time discretization of the Navier-Stokes equations. Remarkably, the fully coupled problem for the solution to the incompressible Stokes equations (combined with the equations that define the smooth extensions to the velocity and pressure fields) can be reduced to the solution of a relatively small[1] dense system of equations, two Stokes solves on a simple domain ignoring the internal boundary, and a handful of fast Fourier transforms. The dense system depends only on the boundary and the discretization, and so it can be formed and prefactored to allow for efficient time-stepping.

[1] With size a small multiple of the number of points used to discretize the boundary

The IBSE method retains the essential robustness and simplicity of the original IB method. All communication between the Lagrangian boundary and the underlying Cartesian grid is achieved by convolution with regularized δ-functions or normal derivatives of those δ-functions. This allows an absolute minimum of geometric information to be used. In the traditional IB method, only the position of the Lagrangian structure must be known; the IBSE method additionally requires normals to that structure and an indicator variable denoting whether Cartesian grid points lie inside or outside of the physical domain where the PDE is defined. As with the IB method, the IBSE method is flexible to the choice of the underlying PDE discretization, so long as the mesh is uniform in the vicinity of the immersed boundary, facilitating simple coupling to existing code. In this paper, we demonstrate the IBSE solver with an underlying Fourier spectral discretization as well as a second-order staggered-grid finite difference discretization.

This paper is organized as follows. In Section 2, we introduce the methodology, ignoring the details of the numerical implementation. This section includes a review of the relevant methodology introduced in [27]. In Section 3, we discuss the numerical implementation of the method, not including the choice of the underlying discretization of the PDE. In Section 4, we study a steady Stokes problem with an analytic solution, using a Fourier spectral discretization. We analyze the convergence of the velocity and pressure fields, as well as derivatives of the velocity fields. We demonstrate up to third-order pointwise convergence of the velocity fields, as well as second-order pointwise convergence of all elements of the stress tensor. In Section 5, we analyze the flow of a viscous incompressible fluid around a cylinder in a confined channel at zero Reynolds number. We solve the problem using both a Fourier spectral discretization and a second-order finite difference discretization, comparing the accuracy and convergence of the solution, as well as the convergence of the scalar drag coefficient to known values from the literature. Finally, in Section 6, we solve the incompressible Navier-Stokes equations in both steady and unsteady cases, for the flow around a cylinder in a confined channel, demonstrating rapid convergence and quantitative agreement with a number of benchmarks from the literature.

2. Methods

We begin by considering the time-dependent, incompressible Navier-Stokes equations:

$$\partial_t \mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{f}_\mathbf{u} \qquad \text{in } \Omega, \tag{1a}$$

$$\nabla\cdot\mathbf{u} = 0 \qquad \text{in } \Omega, \tag{1b}$$

$$\mathbf{u} = \mathbf{b} \qquad \text{on } \Gamma, \tag{1c}$$

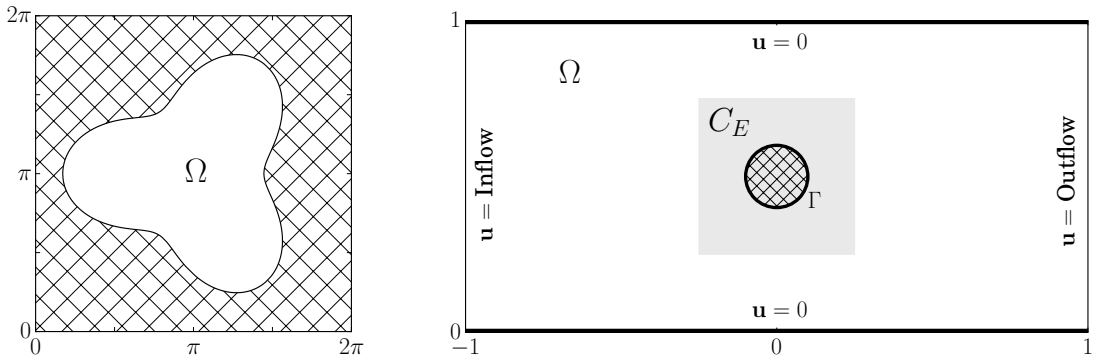
where the boundary of the domain Ω is denoted by $\Gamma = \partial\Omega$. One way to discretize these equations in time is to treat the non-linearity explicitly, and to treat the Stokes operator implicitly. The simplest such time discretization, using forward Euler for the nonlinear terms and backward Euler for the remaining terms, is

$$(\mathbb{I} - \nu\Delta t\Delta)\mathbf{u}^{t+\Delta t} + \Delta t\nabla p^{t+\Delta t} = \mathbf{u}^t + \Delta t\left(\mathbf{f}_\mathbf{u}^{t+\Delta t} - \mathbf{u}^t\cdot\nabla\mathbf{u}^t\right) \qquad \text{in } \Omega, \tag{2a}$$

$$\nabla\cdot\mathbf{u}^{t+\Delta t} = 0 \qquad \text{in } \Omega, \tag{2b}$$

$$\mathbf{u}^{t+\Delta t} = \mathbf{b}^{t+\Delta t} \qquad \text{on } \Gamma. \tag{2c}$$

Although the Navier-Stokes equations are often discretized in time using projection methods [28], these discretizations produce large splitting errors near the boundary at low Reynolds number. Implicit-Explicit time-discretizations of the form of Equation (2) eliminate these errors, and are simpler to generalize to higher-order in time schemes (see Section 6). The unsplit Stokes system may be inverted efficiently using GMRES with a projection preconditioner [29]. We find this inversion strategy to be very efficient (convergence to an algebraic tolerance of 10^{-10} achieved in 3-4 iterations) for moderate to high Reynolds numbers ($\mathrm{Re} \geq 1$), and moderately efficient (convergence achieved in approximately 35 iterations) for the Stokes problem where projection methods cannot be used.



(a) Irregular domain

(b) Flow around cylinder

Figure 1: Two different and typical domains to solve PDE on. The physical domain Ω is shown in white, the extension domain E is denoted by crosshatches. Figure 1a shows an irregular interior domain. The computational domain C is the 2-torus, $\mathbb{T}^2 = [0, 2\pi] \times [0, 2\pi]$. For this domain it is convenient to take $C_E = C$. Figure 1b shows a domain that would be used to compute flow around a cylinder. The computational domain C is taken to be $[-1, 1] \times [0, 1]$, with Dirichlet boundary conditions specified along ∂C (homogeneous for the channel walls at the top and bottom, and specified inflow and outflow conditions at the left and right boundaries of C). The extension is computed on the domain C_E, shown in gray. For this domain C_E does not coincide with C but does contain E. No slip boundary conditions would be imposed on Γ, the interior boundary that separates Ω and E.

Motivated by the form of Equation (2), in this section we focus primarily on developing the methodology required to solve the general Stokes problem:

$$(\alpha\mathbb{I} - \Delta)\mathbf{u} + \nabla p = \mathbf{f_u} \qquad \text{in } \Omega, \tag{3a}$$

$$\nabla \cdot \mathbf{u} = f_p \qquad \text{in } \Omega, \tag{3b}$$

$$\mathbf{u} = \mathbf{b} \qquad \text{on } \Gamma. \tag{3c}$$

When $\alpha = 0$ and $f_p = 0$, Equation (3) reduces to the steady incompressible Stokes equations. The case where $\alpha > 0$ arises in the study of porous media [30] and from discretizing the Navier-Stokes equations in time using an implicit-explicit discretization, as in Equation (2). The non-zero divergence constraint ($f_p \neq 0$) adds no additional complexity and is included for completeness. Before discussing the IBSE method as applied to Equation (3), we first review relevant material from [27], in order to introduce the central ideas in the simplified context of the Poisson problem.

2.1. Review of [27]: the IBSE method for the Poisson problem

Suppose that we wish to solve the Poisson problem

$$\Delta u = f \qquad \text{in } \Omega, \tag{4a}$$

$$u = g \qquad \text{on } \Gamma, \tag{4b}$$

in an arbitrary smooth domain Ω. The IBSE method works by smoothly extending the unknown solution u of a PDE from Ω to a simplier domain C. The domain Ω will be referred to as the *physical domain* and C will be referred to as the *computational domain*. It is assumed that $\Omega \subset C$. The *interior* boundary $\Gamma = \partial\Omega$ will be assumed to be smooth, not self-intersecting, and must separate C into the two disjoint regions Ω and $E = C \setminus \overline{\Omega}$, referred to as the *extension domain*. Neither Ω nor E need be connected. An additional simple computational domain C_E is required to solve the auxiliary equation that defines the extension of the unknown solution. Although C_E may coincide with C, it need only contain E. The boundary of the computational domain C will be denoted by ∂C. Two typical domains are shown in Figure 1. In Figure 1a, C is periodic, and no boundary conditions need to be specified on ∂C. In Figure 1b, Dirichlet conditions are imposed on ∂C. To prevent confusion with the interior boundary Γ, the boundary ∂C and its respective

boundary condition will only be explicitly included in equations when required to prevent confusion, e.g. in Sections 3.4 and 5.2.

We will denote the extension of the unknown solution u by ξ. This extension of the solution is then used to define a volumetric forcing $\mathcal{F}_e = \Delta\xi$ in the region E. With this forcing, an extended problem in all of the simple domain C may be solved:

$$\Delta u_e - \chi_E \mathcal{F}_e = \chi_\Omega f \qquad \text{in } C, \tag{5a}$$

$$u_e = g \qquad \text{on } \Gamma. \tag{5b}$$

The solution u_e gives the desired solution u in Ω and is equal to ξ in E. Because ξ was chosen to be a smooth extension to u, the function u_e is globally smooth in C.

The extension ξ to the unknown function u is defined as a solution to a high-order PDE which takes for its boundary conditions matching criteria of the form $\frac{\partial^j \xi}{\partial n^j} = \frac{\partial^j u}{\partial n^j}$. This allows the extension to be defined by a small number of unknowns (proportional to the number of points used to discretize the boundary). The extension PDE for ξ is solved efficiently in the simple domain C_E using an Immersed Boundary type method.

In order to succinctly describe the methodology we will require some notation. We define the *spread operator*:

$$(S_{(j)}F)(x) = (-1)^j \int_\Gamma F_j(s) \frac{\partial^j \delta(x - X(s))}{\partial n^j}\, dX(s) \tag{6}$$

and the *interpolation operator*:

$$(S^*_{(j)}\xi)(s) = (-1)^j \int_C \xi(x) \frac{\partial^j \delta(x - X(s))}{\partial n^j}\, dx \tag{7}$$

for the j^{th} normal derivative, where $X(s)$ is a parametrization for Γ with s in the parameter interval $\mathcal{I}_\Gamma$. The integral on the right hand side of Equation (7) is notation for the action of the distribution $\frac{\partial^j \delta}{\partial n^j}$ on the smooth function ξ. The traditional Immersed Boundary spread and interpolation operators, $S_{(0)}$ and $S^*_{(0)}$ are denoted by S and S^* respectively. The interpolation operator maps function values (or the normal derivatives of the function) from C to Γ, while the spread operator maps singular (and hyper-singular) forces supported on Γ to C. To simplify equations presented in the forthcoming material, we also define the operators T_k, T^*_k, and R^*_k by:

$$T_k = \sum_{j=0}^{k} S_{(j)}, \tag{8a}$$

$$T^*_k = \begin{pmatrix} S^*_{(0)} & S^*_{(1)} & \cdots & S^*_{(k)} \end{pmatrix}^\intercal, \tag{8b}$$

$$R^*_k = \begin{pmatrix} S^*_{(1)} & \cdots & S^*_{(k)} \end{pmatrix}^\intercal. \tag{8c}$$

The operator T^*_k provides an interpolation of a function and its first k normal derivatives to the boundary; R^*_k provides an interpolation of the first k normal derivatives to the boundary, but excludes the values; the spread operator T_k represents a set of singular forces (δ-like) and hyper-singular forces (like the first k normal derivatives of the δ-function) on the boundary.

The central challenge of the IBSE method is to compute the smooth extension to an *unknown* solution. We first discuss how to compute an extension to a given function. Let $v \in C^k(\Omega)$ be given. To compute a $C^k(C)$ extension to v, we solve the following high-order PDE in the region E:

$$\mathcal{H}^k \xi = 0 \qquad \text{in } E, \tag{9a}$$

$$\frac{\partial^j \xi}{\partial n^j} = \frac{\partial^j v}{\partial n^j} \qquad \text{on } \Gamma,\ 0 \le j \le k. \tag{9b}$$

Here $\mathcal{H}^k$ is an appropriate differential operator such as the polyharmonic operator Δ^{k+1}; we discuss the choice of this operator in more detail, from a numerical perspective, in Section 3.2. This problem may be solved on the simpler domain C_E using methodology directly analogous to the direct forcing Immersed Boundary method. The boundary conditions given in Equation (9b) that force ξ to share its first k normal derivatives with v along Γ are approximated using the operators $S^*_{(j)}$. Unknown singular and hyper-singular forces (F) supported on Γ are regularized and spread to the underlying grid by the operator T_k:

$$\mathcal{H}^k\xi(x) + (T_kF)(x) = 0 \qquad \text{for } x \in C_E, \tag{10a}$$

$$(S^*_{(j)}\xi)(s) = \frac{\partial^j v}{\partial n^j}(s) \qquad \text{for } s \in \mathcal{I}_\Gamma,\ 0 \le j \le k. \tag{10b}$$

The unknown forces F act as Lagrange multipliers enforcing the approximate boundary conditions. Future equations similar to Equation (10b) will be shortened simply to $S^*_{(j)}\xi = \frac{\partial^j v}{\partial n^j}$, and assumed to hold for all $s \in \mathcal{I}_\Gamma$. Notice that ξ is not actually an extension to v: that is, $\xi(x) \neq v(x)$ in Ω. We will only be interested in the function ξ in E, and so need not form its literal extension (which is $\chi_\Omega v + \chi_E \xi$).

To solve the Poisson problem given by Equation (4) using the IBSE method, we instead solve the extended problem given in Equation (5). The forcing function $\mathcal{F}_e$ that is specified in E must be chosen so that it forces the extended solution u_e to be $C^k(C)$. Let ξ smoothly extend u, that is, we ask that ξ is globally smooth in C and that it satisfies the constraints

$$R^*_k\xi = R^*_k u \tag{11}$$

at the interface Γ. These constraints require that the first k normal derivatives of ξ agree with the first k normal derivatives of u on the boundary. Notice that it is not enforced that the *values* of u agree with the values of ξ on Γ; we will see momentarily that this is unnecessary. The forcing function $\mathcal{F}_e$ is then defined as:

$$\mathcal{F}_e = \Delta\xi. \tag{12}$$

Coupling these equations together, we obtain the IBSE formulation for the Poisson problem given by Equation (4):

$$\Delta u_e - \chi_E\Delta\xi = \chi_\Omega f \qquad \text{in } C, \tag{13a}$$

$$\mathcal{H}^k\xi + T_kF = 0 \qquad \text{in } C_E, \tag{13b}$$

$$R^*_k\xi = R^*_k u_e, \tag{13c}$$

$$S^*u_e = 0. \tag{13d}$$

These equations are (13a) the extended Poisson problem, (13b) the extension PDE, (13c) the interface constraints that force u and ξ to share their first k normal derivatives, and (13d) the physical boundary condition to the Poisson problem. Both the interface constraints and the physical boundary condition are enforced by the unknown forces F in the extension PDE. We will subsequently drop the notation u_e and refer to the solution u_e of the IBSE system of equations simply as u.

In Ω, u satisfies the physical Poisson equation; while in E, u satisfies:

$$\Delta u = \Delta\xi \qquad \text{in } E, \tag{14a}$$

$$\frac{\partial u}{\partial n} = \frac{\partial \xi}{\partial n} \qquad \text{on } \Gamma, \tag{14b}$$

and thus $u = \xi$ in E up to a constant. Since $\mathcal{H}^k\xi = -T_kF$ implies that $\xi \in H^{k+1}(C)$, then for $k \ge 1$, the forcing $\chi_\Omega f + \chi_E\Delta\xi \in L^2(C)$, and so $u \in H^2(C)$, and is thus at least $C^0(C)$ (for spatial dimensions $d \le 3$)[2].

[2]The notation $H^l(\Omega)$ denotes the space of measurable functions with the property that the function itself, as well as its first l weak derivatives, are square integrable. The Sobolev embedding theorem implies that any function in $H^2(\Omega)$ is equal almost everywhere to a continuous function, at least in spatial dimensions $d = 2$ and $d = 3$ [31].

This continuity, along with the smoothness constraints guaranteed by the constraint that $R_k^*\xi = R_k^* u$ imply that u is globally $C^k(C)$.

In [27], we verify that the IBSE-k formulation of the problem produces $C^k(C)$ solutions that converge at a rate of $\mathcal{O}(\Delta x^{k+1})$[3] in the grid-spacing Δx across a wide range of Poisson and Poisson-like problems. This is demonstrated for $k = 1$, 2, and 3, corresponding to solutions that converge at a rate of second, third, and fourth order in space for the Poisson problem, and at the same rate in space and fourth-order in time for the heat, Burgers, and the Fitzhugh-Nagumo equations.

Remark 1. In [27], both a volumetric force $\chi_E \Delta\xi$ and a singular force distribution SG are added to the Poisson equation, giving a slightly different formulation for the IBSE method:

$$\Delta u_e - \chi_E \Delta\xi + SG = \chi_\Omega f \qquad \text{in } C, \tag{15a}$$
$$\mathcal{H}^k \xi + T_k F = 0 \qquad \text{in } C_E, \tag{15b}$$
$$T_k^* \xi = T_k^* u_e, \tag{15c}$$
$$S^* u_e = 0. \tag{15d}$$

The singular force distribution SG acts to enforce that the boundary values of ξ equal those of u, this is an unnecessary requirement for the IBSE method. We will proceed with the simpler formulation given in Equation (13) when developing the methodology for the Stokes equation.

2.2. Solution of the general Stokes problem

We now focus on the modifications to the IBSE method required to solve the general Stokes problem:

$$\mathcal{L}\mathbf{u} + \nabla p = \mathbf{f_u} \qquad \text{in } \Omega, \tag{16a}$$
$$\nabla \cdot \mathbf{u} = f_p \qquad \text{in } \Omega, \tag{16b}$$
$$\mathcal{B}(\mathbf{u}, p) = \mathbf{b} \qquad \text{on } \Gamma. \tag{16c}$$

We assume that $\mathbf{f} = (\mathbf{f}_u,\ f_p)$ and $\mathbf{b}$ are sufficiently smooth functions defined on Ω (for $\mathbf{f}$) and Γ (for $\mathbf{b}$). Here $\mathcal{L}$ denotes the Helmholtz operator $\mathcal{L} = \alpha\mathbb{I} - \Delta$ and $\mathcal{B}$ denotes a boundary operator. We discuss the boundary operator associated with the Dirichlet problem ($\mathcal{B}(\mathbf{u}, p) = \mathbf{u}|_\Gamma$); imposition of other boundary conditions is similar and discussed in [27]. Note that this equation reduces to the incompressible Stokes problem when α and f_p are zero.

In the direct forcing Immersed Boundary (IB) method, this problem is solved in a simple computational domain C by adding singular forces $\mathbf{G}$ supported on the boundary that act as Lagrange multipliers which enforce the boundary condition. For Dirichlet problems this can be represented as

$$\mathcal{L}\mathbf{u} + \nabla p + S\mathbf{G} = \mathbf{f_u} \qquad \text{in } C, \tag{17a}$$
$$\nabla \cdot \mathbf{u} = f_p \qquad \text{in } C, \tag{17b}$$
$$S^* \mathbf{u} = \mathbf{b}. \tag{17c}$$

The singular forces supported on Γ induce jumps in the normal derivatives of $\mathbf{u}$ at the boundary; the velocities $\mathbf{u}$ produced by the IB method are generically $C^0(C)$ while the pressure field typically has jump discontinuities. For Eulerian discretizations of the underlying PDE that are ignorant of the boundary, this leads to the slow convergence rate $\mathcal{O}(\Delta x)$ for the velocity field in $L^\infty(\Omega)$. The pressure field p fails to converge near to the boundary, but does converge at the rate $\mathcal{O}(\sqrt{\Delta x})$ in $L^2(\Omega)$.

In order to improve this convergence, the Immersed Boundary Smooth Extension (IBSE) method extends the unknown solutions $\mathbf{u}$ and p to be $C^k(C)$ and $C^{k-1}(C)$, respectively. Suppose that the solution vector

[3] Convergence at the rate of $\mathcal{O}(\Delta x^{k+1})$ is observed for problems with Dirichlet boundary conditions. For Neumann problems, the convergence is $\mathcal{O}(\Delta^k)$[27].

$(\mathbf{u}, p)$ to Equation (16) is known in Ω. Define $\boldsymbol{\xi}_\mathbf{u}$ by

$$\mathcal{H}^k \boldsymbol{\xi}_\mathbf{u} + T_k \mathbf{F}_\mathbf{u} = 0 \qquad \text{in } C_E, \tag{18a}$$
$$R_k^* \boldsymbol{\xi}_\mathbf{u} = R_k^* \mathbf{u}, \tag{18b}$$

and ξ_p by

$$\mathcal{H}^{k-1} \xi_p + T_{k-1} F_p = 0 \qquad \text{in } C_E, \tag{19a}$$
$$T_{k-1}^* \xi_p = T_{k-1}^* p. \tag{19b}$$

Recall that $\mathcal{H}^k$ is a high-order differential operator such as Δ^{k+1}, which will be defined precisely in Section 3.2, and that $\mathbf{F}_\mathbf{u}$ and F_p are unknown singular and hyper-singular forces that act as Lagrange multipliers to enforce the constraints given by Equations (18b) and (19b). We make two remarks concerning the extension of the pressure.

Remark 2. The pressure function p is typically thought of as a Lagrange multiplier: it equals whatever is required to enforce the divergence constraint that $\nabla \cdot \mathbf{u} = 0$. One may suspect therefore that it is unnecessary to construct an explicit extension of the pressure function p as we do in Equation (19). Unfortunately, this is not true. Consider the problem:

$$-\Delta \mathbf{u} + \nabla p = \mathbf{f}_\mathbf{u} \qquad \text{in } \Omega, \tag{20a}$$
$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{20b}$$
$$\mathbf{u} = \mathbf{b} \qquad \text{on } \Gamma, \tag{20c}$$

along with the modified problem where $\mathbf{f}_\mathbf{u}$ is replaced by $\tilde{\mathbf{f}}_\mathbf{u} = \mathbf{f}_\mathbf{u} + \nabla\phi$ for any scalar function ϕ. The pressure will be changed: $\tilde{p} = p - \phi$. But the solution $\mathbf{u}$ will be left unchanged, and thus its extension functions $\boldsymbol{\xi}_\mathbf{u}$ will be unchanged. If the extension of the pressure function depends only locally on $\boldsymbol{\xi}_\mathbf{u}$, it cannot extend both p and $\tilde{p}$ smoothly.

Remark 3. The choice to extend the pressure function p to be $C^{k-1}(C)$ rather than $C^k(C)$ yields improved numerical stability. When $\mathbf{u} \in C^k(C)$, the structure of the Stokes equations implies that $p \in C^{k-1}(C)$. Applying the operator T_k^* to the function p with that global regularity yields an inconsistent estimate of $\frac{\partial^k p}{\partial n^k}$, i.e. $S_{(k)}^* p = \frac{\partial^k p}{\partial n^k} + \mathcal{O}(1)$, and thus enforcing that $T_k^* \xi_p = T_k^* p$ does not actually enforce that $\frac{\partial^k \xi_p}{\partial n^k} = \frac{\partial^k p}{\partial n^k}$.

From these extensions, we may define a set of forces $(\boldsymbol{\mathcal{F}}_\mathbf{u}, \mathcal{F}_p)$:

$$\boldsymbol{\mathcal{F}}_\mathbf{u} = \mathcal{L}\boldsymbol{\xi}_\mathbf{u} + \nabla \xi_p, \tag{21a}$$
$$\mathcal{F}_p = \nabla \cdot \boldsymbol{\xi}_\mathbf{u}. \tag{21b}$$

Analogous to Equation (5), to smoothly extend the Stokes problem given by Equation (16) from Ω to C, we add $\boldsymbol{\mathcal{F}}_\mathbf{u}$ to Equation (16a) and $\boldsymbol{\mathcal{F}}_p$ to Equation (16b) in the extension region E to obtain

$$\mathcal{L}\mathbf{u} + \nabla p = \chi_\Omega \mathbf{f}_\mathbf{u} + \chi_E \mathcal{L}\boldsymbol{\xi}_\mathbf{u} + \chi_E \nabla \xi_p \qquad \text{in } C, \tag{22a}$$
$$\nabla \cdot \mathbf{u} = \chi_\Omega f_p + \chi_E \nabla \cdot \boldsymbol{\xi}_\mathbf{u} \qquad \text{in } C, \tag{22b}$$
$$\mathcal{B}(\mathbf{u}, p) = \mathbf{b} \qquad \text{on } \Gamma. \tag{22c}$$

In the physical domain Ω, we have that

$$\mathcal{L}\mathbf{u} + \nabla p = \mathbf{f}_\mathbf{u} \qquad \text{in } \Omega, \tag{23a}$$
$$\nabla \cdot \mathbf{u} = f_p \qquad \text{in } \Omega, \tag{23b}$$
$$\mathcal{B}(\mathbf{u}, p) = \mathbf{b} \qquad \text{on } \Gamma, \tag{23c}$$

and thus $(\mathbf{u}, p)$ solves the original Stokes problem given in Equation (16). In the extension domain E, we have that

$$\mathcal{L}(\mathbf{u} - \boldsymbol{\xi}_\mathbf{u}) + \nabla(p - \xi_p) = 0 \qquad \text{in } E, \tag{24a}$$

$$\nabla \cdot (\mathbf{u} - \boldsymbol{\xi}_\mathbf{u}) = 0 \qquad \text{in } E, \tag{24b}$$

$$\frac{\partial(\mathbf{u} - \boldsymbol{\xi}_\mathbf{u})}{\partial n} = 0 \qquad \text{on } \Gamma. \tag{24c}$$

Thus $\mathbf{u} = \boldsymbol{\xi}_\mathbf{u}$ and $p = \xi_p$, both up to a constant (in E). Since ξ_p is enforced to match p at the boundary by Equation (19), they are equal at the boundary, and thus $p = \xi_p$ in E, implying that p is globally smooth in C, while the global regularity for $\mathbf{u}$ is implied by the same argument given in Section 2.1. Numerical approximations of these solutions may converge rapidly, even when the PDEs are discretized using an underlying Cartesian grid discretization that is ignorant of the boundary. The difficulty in this formulation comes from the fact that the extensions $\boldsymbol{\xi}_\mathbf{u}$ and ξ_p *depend on the unknown solution* $(\mathbf{u}, p)$.

Combining Equations (16), (18) and (19), we obtain a coupled system for the solution $(\mathbf{u}, p)$ together with the smooth extensions $(\boldsymbol{\xi}_\mathbf{u}, \xi_p)$ to that solution:

Stokes eq.

$$\begin{aligned} \mathcal{L}\mathbf{u} + \nabla p - \chi_E \mathcal{L}\boldsymbol{\xi}_\mathbf{u} - \chi_E \nabla \xi_p &= \chi_\Omega \mathbf{f}_\mathbf{u} && \text{in } C, && (25a)\\ \nabla \cdot \mathbf{u} - \chi_E \nabla \cdot \boldsymbol{\xi}_\mathbf{u} &= \chi_\Omega f_p && \text{in } C, && (25b) \end{aligned}$$

Extension eq.

$$\begin{aligned} \mathcal{H}^k \boldsymbol{\xi}_\mathbf{u} + T_k \mathbf{F}_\mathbf{u} &= 0 && \text{in } C_E, && (25c)\\ \mathcal{H}^{k-1} \xi_p + T_{k-1} F_p &= 0 && \text{in } C_E, && (25d) \end{aligned}$$

Bdy. matching

$$\begin{aligned} R_k^* \boldsymbol{\xi}_\mathbf{u} &= R_k^* \mathbf{u}, && (25e)\\ T_{k-1}^* \xi_p &= T_{k-1}^* p, && (25f) \end{aligned}$$

Phys. Bdy. Cond.

$$S^* \mathbf{u} = \mathbf{b}. \tag{25g}$$

We will refer to this system of equations as the IBSE-k equations. The remainder of this paper is dedicated to demonstrating that these equations provide accurate solutions to the Stokes and Navier-Stokes equations, that they allow a flexible choice of discretization, and that they admit an efficient inversion strategy. We organize the presentation of information as follows:

1. In Section 3, we discuss the details of numerical implementation that do not depend upon the underlying choice of discretization.
2. In Section 4, we empirically select the regularization parameter N introduced in Section 3.2 and solve a Stokes problem with an analytic solution that allows us to carefully verify the convergence rates of the solution $(\mathbf{u}, p)$ (up to third-order for $\mathbf{u}$ and second-order for p, in $L^\infty(\Omega)$), as well as the convergence of elements of the fluid stress tensor σ (up to second-order in $L^\infty(\Omega)$).
3. In Section 5, we describe both a Fourier spectral and a finite-difference discretization to the confined channel flow around a cylinder problem for incompressible Stokes flow. We validate the convergence rates for solutions produced by the IBSE method and compare to known benchmarks.
4. In Section 6, we discuss time-stepping in the IBSE framework, solve both steady and unsteady Navier-Stokes problems, and demonstrate rapid convergence of the solutions and agreement with known benchmarks.

3. Numerical implementation

In this section, we discretize the coupled IBSE-k system given in Equation (25). We assume that the functions are discretized using an underlying Cartesian mesh with uniform[4] grid-spacing Δx but otherwise

[4]For the discretization described here, the mesh needs to be uniform only in a small neighborhood of the boundary, with the same width as the regularized δ-function used to discretize the operators $S_{(j)}$ and $S_{(j)}^*$. In fact, the mesh need not even be

make no assumptions regarding the underlying discretization of the functions and differential operators. These final details will be treated in Section 4 for a Fourier spectral discretization, and in Section 5.2 for a standard second-order finite difference scheme. Little varies in the details.

1. In Section 3.1, we discretize the spread ($S_{(j)}$) and interpolation ($S^*_{(j)}$) operators for the j^{th} normal derivative that were introduced in Equations (6) and (7). Note that the discretization of these operators automatically induces a discretization for the composite operators T_k, T^*_k, and R^*_k, defined in Equation (8).
2. In Section 3.2, we discuss how we choose the extension operator $\mathcal{H}^k$ introduced in Equation (10).
3. In Section 3.3, we describe an efficient inversion strategy for the IBSE-k system given by Equation (25).
4. In Section 3.4, we provide a brief summary of the algorithm.

3.1. Discretization of singular integrals

Let the boundary Γ be parametrized by the function $X(s)$. In all examples in this manuscript, we work with a two-dimensional fluid, and the boundary Γ is one-dimensional and closed, with the single parameter s defined on the periodic interval $[0, 2\pi]$. The discrete version of the spread operator for the j^{th} normal derivative, $S_{(j)} : \Gamma \to C$ defined in Equation (6) requires a regularized δ-function and a discretization of the integral over Γ. The regularized δ-function that we use will be denoted by $\tilde{\delta}$ and is constructed as the Cartesian product of a one-dimensional regularized δ-function denoted by $\tilde{\delta}_1$ that is analytically three times differentiable, satisfies four discrete moment conditions, and has a support width of $16\Delta x$. The $\tilde{\delta}_1$-function and its properties are introduced in [27, 33]. Normal derivatives of $\tilde{\delta}$ are computed by the formula [34]

$$\frac{\partial^j \tilde{\delta}}{\partial n^j} = n_{i_1} \cdots n_{i_j} \frac{\partial^j \tilde{\delta}}{\partial x_{i_1} \cdots \partial x_{i_j}}, \tag{26}$$

where the Einstein summation convention has been used to indicate sums over repeated indices and $\partial^j \tilde{\delta}/\partial x_{i_1} \cdots \partial x_{i_j}$ is computed as Cartesian products of the appropriate derivatives of $\tilde{\delta}_1$. For example, in two dimensions $\partial\tilde{\delta}/\partial n$ is computed as

$$\frac{\partial \tilde{\delta}}{\partial n} = n_x \tilde{\delta}_1' \otimes \tilde{\delta}_1 + n_y \tilde{\delta}_1 \otimes \tilde{\delta}_1', \tag{27}$$

where $\tilde{\delta}_1'$ denotes the first derivative of $\tilde{\delta}_1$.

Discretization of the integral over Γ is made by choosing n_{bdy} quadrature nodes $\tilde{\Gamma} = \{X_i\}_{i=1}^{n_{\text{bdy}}}$, equally spaced in the parameter interval $\mathcal{I}_\Gamma = [0, 2\pi]$ so that $X_i = X(s_i)$ and $s_i = (i-1)2\pi/n_{\text{bdy}}$. Quadrature weights are computed at each quadrature node to be $\Delta s_i = \left\| \frac{\partial X}{\partial s}(s_i) \right\|_2$; this is a spectrally accurate quadrature rule for the integral of smooth periodic functions on Γ. The discrete spread operator $S_{(j)}$ maps functions sampled at points in $\tilde{\Gamma}$ to C by

$$(S_{(j)}F)(x) = \sum_{i=1}^{n_{\text{bdy}}} F(s_i) \frac{\partial^j \tilde{\delta}(x - X_i)}{\partial n^j} \Delta s_i. \tag{28}$$

We do not adopt explicit notation to distinguish between the analytic and discrete operators. The number of points in the quadrature is chosen so that $\Delta s \approx 2\Delta x$. This choice of node-spacing is wider than that recommended for the traditional IB method [1] but has been observed empirically to be the optimal choice in other studies of *direct-forcing* IB methods [4]. We define the interpolation operator $S^*_{(j)}$ by the adjoint property $\left\langle u, S_{(j)}F \right\rangle_C = \left\langle S^*_{(j)}u, F \right\rangle_\Gamma$, but note that the discrete interpolation operator $S^*_{(j)}$ produces a discrete function

$$(S^*_{(j)}u)(s_k) = \int_C u(x) \frac{\partial^j \tilde{\delta}(x - X_k)}{\partial n^j}\, dx. \tag{29}$$

uniform, but this would add some complexity to the discretization [32].

Discrete integrals over C are straightforward sums computed over the underlying uniform Cartesian mesh, and may be efficiently evaluated due to the finite support of $\tilde{\delta}$. When acting on vector functions, all operators defined in this section are assumed to operate elementwise.

3.2. Solution of the extension equations and the choice of extension operator $\mathcal{H}^k$

Due to the difficulty in accurately and efficiently inverting the high-order differential operator needed to define the smooth extensions, we solve these equations utilizing a Fourier spectral discretization regardless of the discretization used for the Stokes equations. Let m be the largest wave-number associated with the discrete Fourier transform (DFT) on the discretized domain C_E. We choose the extension operator $\mathcal{H}^k$ introduced in Equation (9) to be:

$$\mathcal{H}^k = \Delta^{k+1} + (-1)^{k+1}\Theta(k, m). \tag{30}$$

Here Θ is a positive scalar function that depends on the smoothness of the extension (k) and the largest wave-number (m). The function Θ is chosen to mitigate the numerical condition number of the operator $\mathcal{H}^k$:

$$\kappa = 1 + \frac{m^{2(k+1)}}{\Theta}. \tag{31}$$

Minimizing the condition number κ (by taking Θ to be large) must be balanced against the need to resolve the intrinsic length scale $L = \Theta^{-1/2(k+1)}$ introduced to the problem. We set the length scale L as a function of Δx, in effect allowing the extension to decay to zero over some number of grid cells. We thus choose

$$\Theta^* = \left(\frac{1}{N\Delta x}\right)^{2(k+1)}, \tag{32}$$

where N is proportional to the number of gridpoints over which the extension decays. The choice of N can impact the accuracy of the scheme: if N is too large, the condition number of $\mathcal{H}^k$ may be high with respect to the precision of the computer being used, and numerical instability causes a degradation of the quality of the solution. If N is too small, short length scales that are not resolvable by the discretization are introduced to the problem. For all examples presented in this paper, N is chosen to be 200. In Section 4, we investigate the effect of the choice of N on a specific problem.

3.3. Inversion of the IBSE-k system, Equation (25)

In this section, we describe an efficient inversion strategy for Equation (25). We will assume that the discrete Stokes operator is invertible[5]. The details for when it is not invertible (e.g. when solving on a periodic domain with $\mathcal{L} = -\Delta$) are analogous to the Poisson problem, and the solution is presented in [27]. In matrix form, Equation (25) is given by:

$$\left(\begin{array}{cccc|cc}
\mathcal{L} & \nabla\cdot & -\chi_E\mathcal{L} & -\chi_E\nabla & & \\
\nabla\cdot & & -\chi_E\nabla\cdot & & & \\
 & & \mathcal{H}^k & & T_k & \\
 & & & \mathcal{H}^{k-1} & & T_{k-1} \\
\hline
S^* & & & & & \\
R_k^* & & -R_k^* & & & \\
 & T_{k-1}^* & & -T_{k-1}^* & & \\
\end{array}\right)
\left(\begin{array}{c}
\mathbf{u} \\ p \\ \boldsymbol{\xi}_u \\ \xi_p \\ \hline \mathbf{F_u} \\ F_p
\end{array}\right)
=
\left(\begin{array}{c}
\chi_\Omega \mathbf{f_u} \\ \chi_\Omega f_p \\ 0 \\ 0 \\ \hline \mathbf{b} \\ 0 \\ 0
\end{array}\right). \tag{33}$$

This system is large, but the number of constraints (the last three equations) is relatively small. By block Gaussian elimination, we can find a Schur-complement for this matrix, which we label SC:

$$SC = \begin{pmatrix} S^* & & & \\ R_k^* & & -R_k^* & \\ & T_{k-1}^* & & -T_{k-1}^* \end{pmatrix}
\begin{pmatrix} \mathcal{L} & \nabla\cdot & -\chi_E\mathcal{L} & -\chi_E\nabla \\ \nabla\cdot & & -\chi_E\nabla\cdot & \\ & & \mathcal{H}^k & \\ & & & \mathcal{H}^{k-1} \end{pmatrix}^{-1}
\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ T_k & 0 \\ 0 & T_{k-1} \end{pmatrix} \tag{34}$$

[5]The null-space in the pressure function p should be fixed by the discrete operator; e.g. by enforcing that $\int_C p = 0$.

along with an associated system of equations for the Lagrange multipliers $\mathbf{F_u}$ and F_p,

$$\begin{pmatrix} & SC & \end{pmatrix} \begin{pmatrix} \mathbf{F_u} \\ F_p \end{pmatrix} = \begin{pmatrix} S^* & & \\ R_k^* & & \\ & & T_{k-1}^* \end{pmatrix} \begin{pmatrix} \mathcal{L} & \nabla \\ \nabla\cdot & \end{pmatrix}^{-1} \begin{pmatrix} \chi_\Omega \mathbf{f_u} \\ \chi_\Omega f_p \end{pmatrix} - \begin{pmatrix} \mathbf{b} \\ 0 \\ 0 \end{pmatrix}. \tag{35}$$

The size of SC is comparatively small, only $(2(k+1)+k)n_{\text{bdy}}$ square. This equation is the key to the efficiency of the algorithm, as it allows the Lagrange multipliers $\mathbf{F_u}$ and F_p to be computed rapidly without first solving for $\mathbf{u}$, p, $\boldsymbol{\xi}_\mathbf{u}$, or ξ_p. Once $\mathbf{F_u}$ and F_p are determined, $\boldsymbol{\xi}_\mathbf{u}$ and ξ_p may be found by solving Equations (25c) and (25d). With $\boldsymbol{\xi}_\mathbf{u}$ and ξ_p known, $\mathbf{u}$ and p may be computed simply by solving the Stokes equation in C, ignoring the boundary Γ (that is, by solving Equations (25a) and (25b)).

Rapid inversion of the system of equations for $\mathbf{F_u}$ and F_p given in Equation (35) is not a trivial task. Because we have restricted to problems set on stationary domains and the size of SC is small, it is feasible to proceed using dense linear algebra. We form SC by repeatedly applying it to basis vectors. This operation is expensive: for two dimensions it is $\mathcal{O}(N^{3/2}\log N)$ in the total number of unknowns $N = n^2$. The Schur-complement is then factored by the LU algorithm provided by LAPACK [35]. Once this factorization is computed Equation (35) can be solved rapidly. This Schur-complement depends only on the domain and the discretization, so the LU-decomposition can be reused to solve multiple problems on the same domain or in each timestep when solving time-dependent problems.

3.4. Outline of the methodology

We provide a brief outline of the steps required for solving the general Stokes problem given by Equation (16) using the IBSE-k method. The algorithm proceeds as follows:

1. Form the right hand side of the Schur-complement equation, as defined in Equation (35). This may be done by solving

$$\begin{aligned} \mathcal{L}\mathbf{u}_0 + \nabla p_0 &= \chi_\Omega \mathbf{f_u} && \text{in } C, && \text{(36a)} \\ \nabla\cdot\mathbf{u}_0 &= \chi_\Omega f_p && \text{in } C, && \text{(36b)} \\ \mathcal{B}(\mathbf{u}_0, p_0) &= \mathbf{b}_C && \text{on } \partial C, && \text{(36c)} \end{aligned}$$

   for $(\mathbf{u}_0, p_0)$ and then computing $S^*\mathbf{u}_0 - \mathbf{b}$, $R_k^*\mathbf{u}_0$ and $T_{k-1}^* p_0$. Note that the *internal* boundary Γ is ignored when computing the solution to Equation (36).
2. Compute $\mathbf{F_u}$ and F_p by solving the Schur-complement system given by Equation (35), given the right hand side computed above. We assume that the Schur complement SC has been formed and its LU decomposition found as a pre-computation.
3. Compute $\boldsymbol{\xi}_\mathbf{u}$ and ξ_p from $\mathbf{F_u}$ and F_p by solving Equations (25c) and (25d). We take C_E to always have periodic boundary conditions, and these equations may be simply inverted using the fast Fourier transform.
4. With $\boldsymbol{\xi}_\mathbf{u}$ and ξ_p known, find $\mathbf{u}$ and p by solving Equations (25a) and (25b). To be precise, we solve

$$\begin{aligned} \mathcal{L}\mathbf{u} + \nabla p &= \chi_E \mathcal{L}\boldsymbol{\xi}_\mathbf{u} + \chi_E \nabla \xi_p + \chi_\Omega \mathbf{f_u} && \text{in } C, && \text{(37a)} \\ \nabla\cdot\mathbf{u} &= \chi_E \nabla\cdot\boldsymbol{\xi}_\mathbf{u} + \chi_\Omega f_p && \text{in } C, && \text{(37b)} \\ \mathcal{B}(\mathbf{u}, p) &= \mathbf{b}_C && \text{on } \partial C. && \text{(37c)} \end{aligned}$$

   Note again that the *internal* boundary Γ is ignored when computing this solution.

Step two of this algorithm requires that the Schur-complement SC be formed and factored as a pre-processing step. The Schur complement operator maps a set of singular (and hyper-singular) forces $(\mathbf{F_u}, F_p)$ to the constraints given by Equations (25e) to (25g). We form the matrix by applying SC to basis vectors.

To apply SC to any set of singular forces, first compute $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p from $\mathbf{F}_u$ and F_p by solving Equations (25c) and (25d). Then $\mathbf{u}$ and p may be found by solving

$$\begin{aligned} \mathcal{L}\mathbf{u} + \nabla p &= \chi_E \mathcal{L}\boldsymbol{\xi}_{\mathbf{u}} + \chi_E \nabla \xi_p && \text{in } C, && (38a)\\ \nabla \cdot \mathbf{u} &= \chi_E \nabla \cdot \boldsymbol{\xi}_{\mathbf{u}} && \text{in } C, && (38b)\\ \mathcal{B}(\mathbf{u}, p) &= 0 && \text{on } \partial C, && (38c) \end{aligned}$$

and finally with $\mathbf{u}$, p, $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p known, then $S^*\mathbf{u}$, $R_k^*\mathbf{u} - R_k^*\boldsymbol{\xi}_{\mathbf{u}}$, and $T_{k-1}^* p - T_{k-1}^* \xi_p$ can be evaluated. Note that $\mathbf{f}_{\mathbf{u}}$, f_p and $\mathbf{b}_C$ do not enter into this computation. The Schur complement SC depends only on the geometry and the discrete operators $\mathcal{L}$, ∇, $\nabla\cdot$.

4. Results: Stokes equation with analytic solution

To demonstrate the improved convergence properties of the IBSE method as compared to the IB method, we construct a simple example with an analytic solution and compare the accuracy of the solutions with those produced by the Immersed Boundary method. For the numerical comparisons with the IB method provided in this section, we show two sets of values: those produced using a regularized δ-function commonly used in IB simulations [10]:

$$8\,\delta(x) = \begin{cases} 3 - 2|x| + \sqrt{1 + 4|x| - 4|x|^2} & 0 \le x \le 1, \\ 5 - 2|x| - \sqrt{-7 + 12|x| - 4|x|^2} & 1 < x \le 2, \\ 0 & 2 < x, \end{cases} \tag{39}$$

as well as those produced using the smoother and more accurate δ function introduced in [27] and used in all IBSE simulations in this paper. These two methods will be denoted by $\mathrm{IB}_{\sqrt{\cdot}}$ and IB_4, respectively. We solve the general Stokes problem with $\mathcal{L} = \mathbb{I} - \Delta$ and Dirichlet boundary conditions:

$$\begin{aligned} \mathbf{u} - \Delta\mathbf{u} + \nabla p &= \mathbf{f}_{\mathbf{u}} && \text{in } \Omega, && (40a)\\ \nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega, && (40b)\\ \mathbf{u} &= \mathbf{b} && \text{on } \Gamma, && (40c) \end{aligned}$$

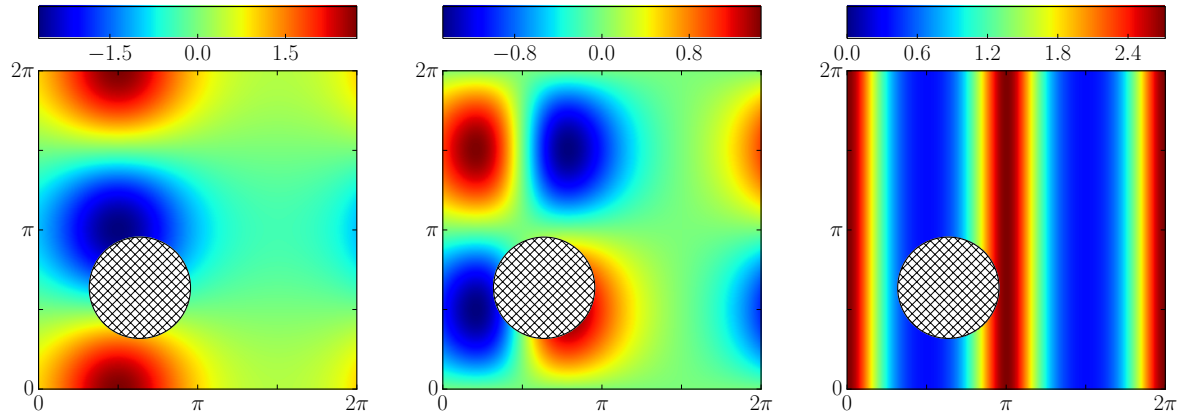
where the solution $(\mathbf{u}, p)$ is given by

$$u(x,y) = e^{\sin x}\cos y; \quad v(x,y) = -\cos x\, e^{\sin x}\sin y; \quad p(x,y) = e^{\cos 2x}, \tag{41}$$

the forcing function $\mathbf{f}_{\mathbf{u}}$ is defined as $\mathbf{f}_{\mathbf{u}} = \chi_\Omega\,(\mathbf{u} - \Delta\mathbf{u} + \nabla p)$, and the boundary condition $\mathbf{b}$ is found by evaluating $\mathbf{u}$ on Γ. The physical domain Ω is taken to be $\mathbb{T}^2 \setminus \overline{B_1(2,2)}$, where $\mathbb{T}^2$ denotes the 2-torus identified with the periodic rectangle $[0, 2\pi] \times [0, 2\pi]$, and where $B_1(2,2)$ denotes the ball of radius 1 centered at $(2,2)$. The computational domain is taken to be $C = \mathbb{T}^2$. Plots of the analytic solution $(\mathbf{u}, p)$ in the physical domain Ω are shown in Figure 2. For this computational domain it is convenient to use a Fourier spectral discretization for C and to choose $C_E = C$. We discretize the domain $[0, 2\pi] \times [0, 2\pi]$ using a regular Cartesian mesh with n points discretizing each dimension: $\Delta x = 2\pi/n$ and $x_j = j\Delta x$ for $0 \le j < n$. Differential operators are discretized in the usual way.

4.1. Determination of the regularization parameter N

Before analyzing the convergence of solutions produced by the IBSE method, we investigate the choice of N, the parameter used to regularize the extension operator $\mathcal{H}^k$, as introduced in Equation (32). This parameter controls the rate at which the extension functions $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p decay to 0 away from the boundary. To choose N, we compute solutions with $n = 256$, using extensions across a wide range of choices of N (from $N = 1$ to $N = 40000$), and for $k = 1$, $k = 2$, and $k = 3$. The $L^\infty(\Omega)$ error for u, and the condition number of the Schur complement, as a function of N, are shown in Figures 3a and 3b. In Figures 3c and 3d, we show the extension function ξ_u for the solution u to Equation (40) for $N = 1$ and $N = 200$.

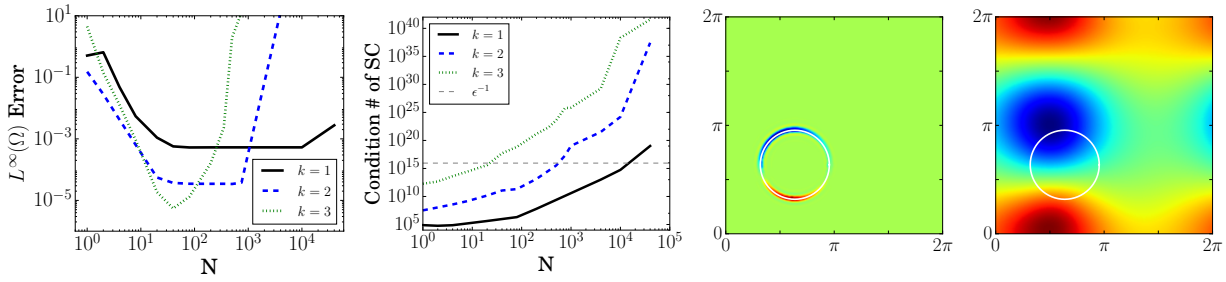


(a) u (b) v (c) p

Figure 2: The analytic solution $(\mathbf{u}, p)$ and the physical domain Ω to the problem defined by Equations (40) and (41). The extension domain E is denoted with crosshatches.

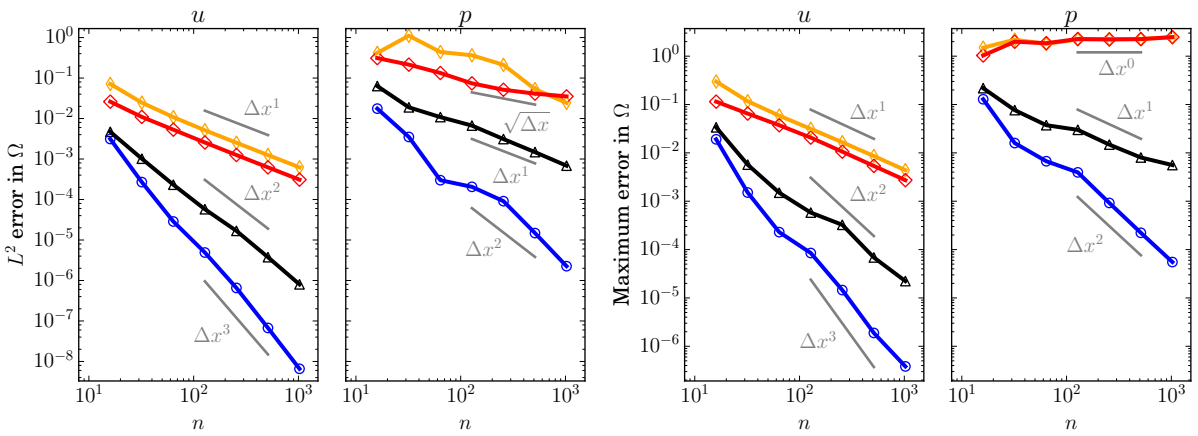
For small values of N, the error is dominated by the inability of the discretization to resolve the small length scale introduced to the problem by the operator $\mathcal{H}^k$. As N is increased, the error produced by the IBSE method improves, but the condition number of the Schur-complement worsens. A steep increase in error is observed at approximately the same value of N for which the condition number of the Schur complement matrix exceeds the inverse of machine-epsilon, shown in gray in Figure 3b for a double precision machine. For $k = 1$ and $k = 2$, there is a flat region where the error is dominated by the error due to truncation of the Fourier series for the solution to the PDE coupled with its smooth extension. For $k = 1$, this region is approximately $50 \leq N \leq 10000$; for $k = 2$ this region is approximately $100 \leq N \leq 750$. Other than for this test, all numerical results in this paper are produced using $N = 200$ to construct $\mathcal{H}^k$. Similar tests (across a coarser set of values of N) have been done for all test problems in the paper and across a range of values of n; the value $N = 200$ provides optimal or nearly-optimal error in all cases. For $k = 3$, no value of N provides optimal results: the error is dominated either by the failure to resolve the artificial length scale introduced by $\mathcal{H}^k$ or by the ill-conditioning of the Schur-complement matrix. Because of this, we have shown no results for IBSE-3[6]. With the choice of N made, we can now analyze the spatial convergence of

[6]For Poisson and Poisson-like problems, the conditioning problems are less severe, and the IBSE-3 method is able to achieve accurate and stable solutions that converge at $\mathcal{O}(\Delta x^4)$ [27].



(a) $L^\infty(\Omega)$ error in u (b) Condition # of SC (c) $N = 1$ (d) $N = 200$

Figure 3: Figures 3a and 3b show the $L^\infty(\Omega)$ error in u for the solution to Equation (40) and the condition number of the Schur Complement defined by Equation (34) produced using different values of N. Figures 3c and 3d show the extension function ξ_u to u, with $n = 256$ and $k = 2$, for $N = 1$ and $N = 200$ (the value used in all simulations in this paper, other than this test). When N is too small (e.g. $N = 1$) short length scales not fully resolvable by the discretization are introduced to the problem.



(a) $L^2(\Omega)$ Error (b) $L^\infty(\Omega)$ Error

Figure 4: $L^2(\Omega)$ and $L^\infty(\Omega)$ errors for the solution to Equation (40) for the $\text{IB}_{\sqrt{\cdot}}$(◊), IB_4(◇), IBSE-1 (△), and IBSE-2 (○) methods.

the IBSE-1 and IBSE-2 methods.

4.2. Convergence analysis

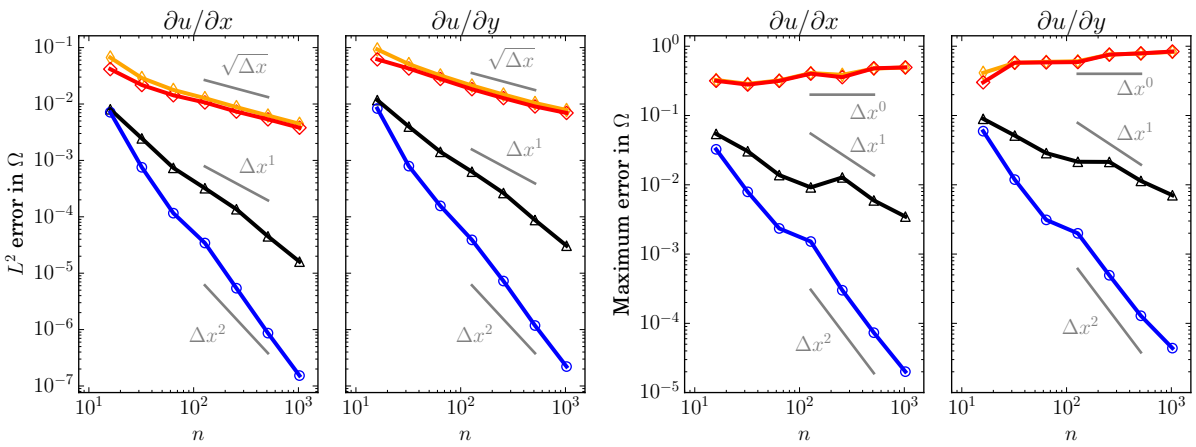
A refinement study for the $L^2(\Omega)$ and $L^\infty(\Omega)$ errors in the solutions for u and p is shown in Figure 4. The refinement path for v is omitted; it is similar to what is shown for u. In $L^2(\Omega)$, the velocity field u converges at first-, second-, and third-order accuracy in Δx for the IB, IBSE-1 and IBSE-2 methods, respectively, while the pressure field p converges at first- and second-order accuracy for IBSE-1 and IBSE-2, but at the slow rate of $\mathcal{O}(\sqrt{\Delta x})$ for the IB method. In $L^\infty(\Omega)$, the rates of convergence for the velocity remain unchanged in all methods, while for the pressure the difference is starker: the IBSE-1 and IBSE-2 methods maintain first- and second-order convergence, while the IB method *fails to converge pointwise.*

Remark 4. It may be surprising that the velocity fields produced by the Immersed Boundary method converge at only first-order in $L^2(\Omega)$. Indeed, for a given $\mathbf{F}$ and $\mathbf{b}$, the velocity fields produced by solving

$$-\Delta\mathbf{u} + \nabla p + S\mathbf{F} = 0, \tag{42a}$$
$$\nabla \cdot \mathbf{u} = 0, \tag{42b}$$
$$S^*\mathbf{u} = \mathbf{b}, \tag{42c}$$

converge at $\mathcal{O}(\Delta x)$, $\mathcal{O}(\Delta x^{3/2})$, and $\mathcal{O}(\Delta x^2)$ in the $L^\infty(\Omega)$, $L^2(\Omega)$, and $L^1(\Omega)$ norms, respectively [36]. However, $\mathbf{F}$ is not given, it is computed by inverting a Schur-complement equation whose elements include terms of the form $S^*\mathbf{u}$ where $\mathbf{u} \in C^0(C)$, and thus contain $\mathcal{O}(\Delta x)$ errors.

In the refinement study shown in Figure 4, the pressure function converges more slowly than the velocity function; in the case of the IB method, it fails to converge pointwise. This will be true in general for elements of the stress tensor, which are one derivative less smooth than the velocity field. In Figure 5, we show a refinement study for the $L^2(\Omega)$ and $L^\infty(\Omega)$ errors in the derivatives $\partial u/\partial x$ and $\partial u/\partial y$ (the convergence paths for $\partial v/\partial x$ and $\partial v/\partial y$ are omitted, but essentially the same). A similar pattern to the convergence of the pressure function is observed: for all of the derivatives, the IB method converges slowly (at $\mathcal{O}(\sqrt{x})$) in $L^2(\Omega)$, but fails to converge pointwise; the IBSE-1 and IBSE-2 method provide first- and second-order convergence in both $L^\infty(\Omega)$ and $L^2(\Omega)$. We emphasize that with pointwise convergence of the pressure function p as well as pointwise convergence of these derivatives, the IBSE method is able to capture the stress tensor $\sigma = \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^\intercal) - p\mathbb{I}$ *pointwise up to the boundary* Γ.



(a) $L^2(\Omega)$ Error (b) $L^\infty(\Omega)$ Error

Figure 5: $L^2(\Omega)$ and $L^\infty(\Omega)$ errors for the derivatives of the velocity fields that solve Equation (40), as computed by the $\text{IB}_{\sqrt{\cdot}}$(◊), IB_4(◇), IBSE-1 (△), and IBSE-2 (○) methods.

Remark 5. In this section, we presented results for the Immersed Boundary method using both the $\text{IB}_{\sqrt{\cdot}}$ and IB_4 functions to discretize the S and S^* operators. For all results presented in this section, we observe the same asymptotic behavior, although the IB_4 method typically provides lower errors. For the remainder of this paper, we will present results for the Immersed Boundary method using only the more commonly used $\text{IB}_{\sqrt{\cdot}}$ function.

5. Stokes flow around a cylinder

In this section we focus on the zero Reynolds number flow around a cylinder in a confined channel. The exact problem we will simulate is on the domain $[-20, 20] \times [-2, 2]$, with a cylinder with radius $R = 1$ centered at $(0, 0)$, and a mean stream-wise flow velocity of $u = 1$ enforced at the inflow boundary (where $x = -20$). No-slip ($\mathbf{u} = 0$) boundary conditions are imposed on the channel and cylinder walls. We choose this setup as this domain has been extensively studied in the context of polymeric flow problems [37–39], and accurate benchmarks are available.

We solve this problem using two separate underlying discretizations. In Section 5.1, we describe a Fourier spectral discretization. The Fourier discretization allows the method to achieve third-order accuracy for the velocity field $\mathbf{u}$ when C^2 extensions are used, but when using this discretization, the enforcement of the no-slip $\mathbf{u} = 0$ boundary condition along the long channel walls requires the use of the IBSE methodology. This leads to a large number of boundary points n_{bdy} and a very large Schur-complement operator that must be formed and inverted.

To help alleviate this problem, in Section 5.2 we couple the IBSE method to a standard second-order staggered-grid finite difference discretization of the Stokes equations. The no-slip condition on the channel walls is enforced naturally by the discretization, and the IBSE methodology is used only to enforce the no-slip condition along the cylinder wall and to smoothly extend the unknown solution to the interior of the cylinder. Despite the discretization being limited to second-order accuracy, we show that there is still a benefit to using the IBSE-2 method over the IBSE-1 method, especially when local information regarding the stress is required. If higher-order accuracy is desired, a more accurate finite-difference discretization could be employed.

In Section 5.3, we compare the accuracy of these two different discretizations, for the $\text{IB}_{\sqrt{\cdot}}$, IBSE-1, and IBSE-2 methods, and finally in Section 5.4, we compare the numerical efficiency of the different discretizations.

5.1. Fourier spectral discretization

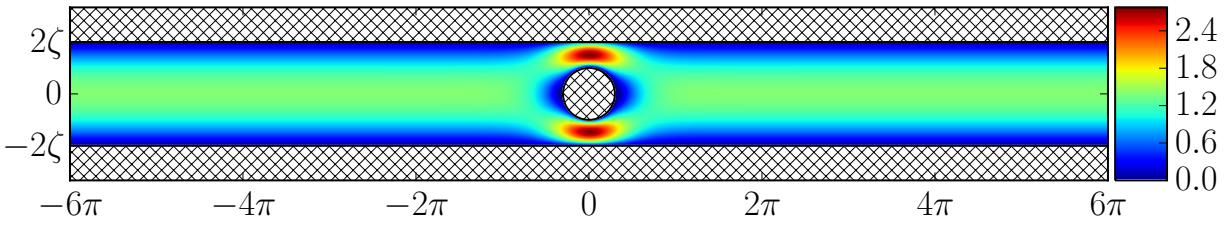


Figure 6: The streamwise velocity u and solution domain for the Fourier spectral discretization to the Stokes channel flow around a cylinder problem defined in Section 5.1. The extension domain E is denoted with crosshatches. The solution that is shown was produced using the IBSE-2 method with $n_y = 512$. Note that space is rescaled and ζ is chosen so that this domain is equivalent to the $[-20, 20] \times [-2, 2]$ domain used in other studies.

For this problem we take the computational domain to be $C = [-6\pi, 6\pi] \times [-\pi, \pi]$, with channel walls located at $y = \pm 2\zeta$, where $\zeta = 12\pi/40$; and the cylinder of radius $R = \zeta$ is centered at $(0, 0)$. This domain, together with the streamwise velocity u produced using the IBSE-2 method with $n_y = 512$, is shown in Figure 6. The domain is discretized with a uniform Cartesian mesh, with n_y points in the spanwise direction and $n_x = 6n_y$ points in the streamwise direction, with $\Delta x = 2\pi/n_y$. The discrete meshpoints x_{ij} are located at $x_{ij} = (-6\pi + i\Delta x, -\pi + j\Delta x)$ for $0 \le i < n_x$ and $0 \le j < n_y$. All differential operators are discretized in the usual way, and the spread and interpolation operators are discretized as described in Section 3.1. Because the extension region E (denoted by crosshatches in Figure 6) is not localized to a small region of C, we choose the domain on which the extension equations are solved to coincide with C, i.e. $C_E = C$. In order to simulate a (rescaled) Poiseuille inflow condition, we solve the problem:

$$-\Delta \mathbf{u} + \nabla p = \beta \hat{x} \qquad \text{in } \Omega, \tag{43a}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{43b}$$

$$\mathbf{u} = 0 \qquad \text{on } \Gamma, \tag{43c}$$

$$\frac{1}{4\zeta} \int_{-2\zeta}^{2\zeta} u(-6\pi, y)\, dy = \zeta, \tag{43d}$$

where Γ denotes the internal boundaries (the surfaces of the cylinder and channel walls). The magnitude of the constant forcing β is treated as a Lagrange multiplier, and is used to enforce the constraint that the spanwise average of the streamwise velocity (u) at the far left end of the domain is ζ. The integral across the channel in Equation (43d) is discretized using Simpson's rule. In [37–39], the boundary condition specified at the inflow is that $u(-20, y) = 3(4 - y)^2/8$. Our computational setup approximates this boundary condition but does not impose it exactly. Given the length of the channel compared to R, we expect our solutions to match closely with other studies; results are shown in Section 5.3.

5.2. Second-order finite difference discretization

In this section we describe how to couple the IBSE method to a standard finite difference scheme. The Stokes Equations (16) are discretized using a second-order finite difference staggered grid discretization where the velocity unknowns (u, v) are stored on the vertical and horizontal cell edges, respectively, and the pressure unknowns p are stored at the cell centers [40, 41]. For this discretization, the computational domain is taken to be $C = [-20, 20] \times [-4, 4]$. There is no need to provide an extension region along the channel walls, as in the domain for the Fourier spectral discretization shown in Figure 6. No-slip conditions ($\mathbf{u} = 0$) are applied on the top and bottom walls at $y = -2$ and $y = 2$; an inflow condition of $\mathbf{u} = (3(4 - y^2)/8, 0)$ is applied at $x = -20$; and an approximate outflow condition of $\partial \mathbf{u}/\partial x$ is applied at $x = 20$. Note that for this discretization, the flow is driven by the boundary condition, and no additional forcing has to be added to the momentum equation as in Equation (43). Inversion of the discrete Stokes system (Equation (36))

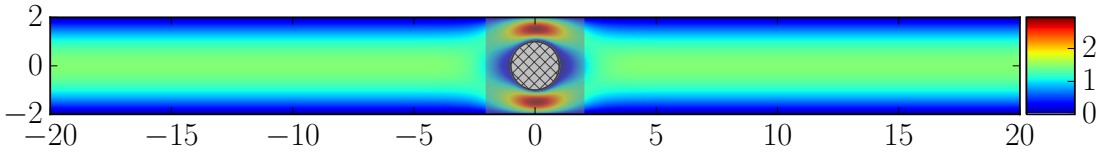


Figure 7: The solution u to the confined channel flow around a cylinder, produced using a second-order finite difference discretization to the Stokes equations as the underlying PDE solver for the IBSE-2 method with $n_y = 512$, as described in Section 5.2. The extension region E is denoted by crosshatches. The extension functions $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p are not solved for on the entirety of the computational domain, but rather in the much smaller region $C_E = [-2, 2] \times [-2, 2]$, localized around E, that is shaded in light gray.

is accomplished using a preconditioned GMRES method with a projection preconditioner [42]. With the operator $\mathcal{L} = \theta\mathbb{I} - \nu\Delta_{\mathbf{u}}$, this preconditioner is given by

$$P^{-1} = \begin{pmatrix} \mathbb{I} & \nabla\Delta_p^{-1} \\ 0 & -\theta\Delta_p^{-1} + 2\nu\mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ -\nabla\cdot & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathcal{L}^{-1} & 0 \\ 0 & \mathbb{I} \end{pmatrix}. \tag{44}$$

The discrete operators $\mathcal{L}$ and Δ_p are inverted exactly using a combination of fast sine/cosine transforms in one direction and tri-diagonal matrix inverses in the other direction. In our numerical tests this preconditioner provides convergence to a relative residual of 10^{-10}, independent of the grid spacing Δx, in approximately 30-40 iterations for the Stokes problem studied in this section, and approximately 3-4 iterations for the Navier-Stokes problems studied in Section 6.1 and Section 6.2.

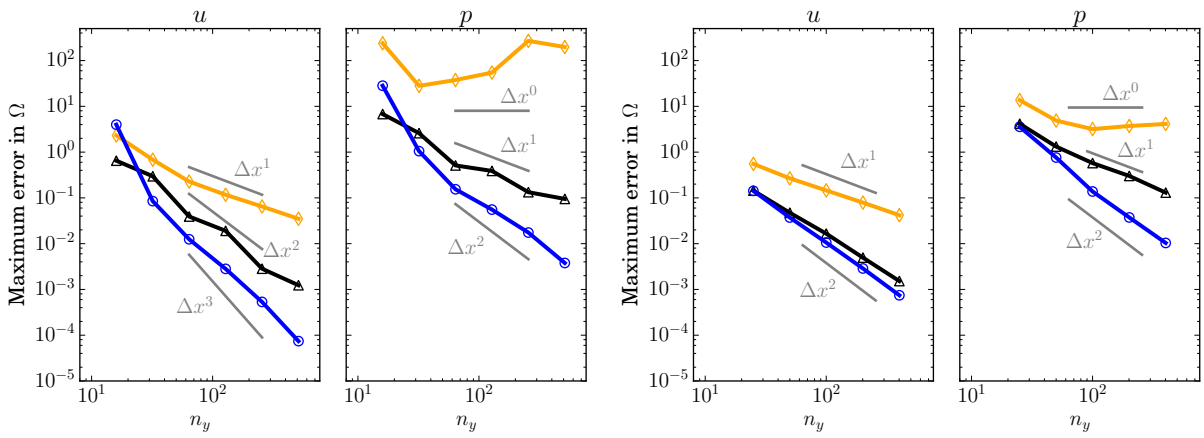
In principle, coupling the IBSE method to this discretization is simple. Examining the outline given in Section 3.4, the only details that are not fully specified are the choice of C_E (the computational domain for the extension equations), and the rules for transferring $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p from the grid or grids on which the extensions $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p are solved on to the staggered grids for $(\mathbf{u}, p)$.

In Figure 7, we show the domain C for this problem, together with the solution computed by the IBSE-2 method with $n_y = 512$. Notice that the extension region E, denoted by crosshatches, is isolated to a small region of C. Solving the extension equations for $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p on all of C would be computationally wasteful. Instead, C_E is chosen for this problem to be $[-2, 2] \times [-2, 2]$, which is highlighted in gray in Figure 7. In principle, the only requirement for C_E is that it contain E; in practice, it should be chosen large enough so that the periodic boundary conditions used when solving the extension equation on C_E do not induce any undesirably large gradients. The operators $\mathcal{H}^k$ and $\mathcal{H}^{k-1}$ are discretized and inverted using Fourier spectral methods.

Because the staggered discretization we use does not collocate the unknowns for the velocity fields u and v and the pressure field p, some choice must be made as to where the unknowns for the extension functions ξ_u, ξ_v, and ξ_p should be located. We have explored two options:

1. A single extension 'grid' C_E, where ξ_u, ξ_v, and ξ_p are collocated. Bilinear interpolation is used to transfer information between this grid and the staggered grids for u, v, and p.
2. Three extension grids ${C_E}^u$, ${C_E}^v$, and ${C_E}^p$, on which the nodes for ξ_u, ξ_v, and ξ_p are each collocated with the nodes for u, v, and p, respectively. No interpolation is necessary when transferring information between grids. The extension grid ${C_E}^u = [0, 2] \times [\Delta x/2, 2 + \Delta x/2]$ is shifted from $[0, 2] \times [0, 2]$ by $\Delta x/2$ in the y direction so that the locations of the unknowns coincide with the locations of the unknowns for the u grid; the extension grid ${C_E}^v$ is shifted by $\Delta x/2$ in the x direction, and the extension grid ${C_E}^p$ is shifted by $\Delta x/2$ in both the x and y directions.

In our numerical experiments, the second option produced Schur-complement operators with slightly lower numerical condition numbers and improved stability when solving time-dependent equations. Although this choice requires slightly greater computational complexity, we use this option for this problem as well as the Navier-Stokes problems defined in Section 6.



(a) Spectral

(b) Finite difference

Figure 8: Refinement study showing the convergence of differences between successively refined solutions, in $L^\infty(\Omega)$, for the solutions u and p to the Stokes confined channel flow around a cylinder problem, computed by the $\text{IB}_{\sqrt{\cdot}}$(◊), IBSE-1 (△), and IBSE-2 (○) methods.. Figure 8a shows solutions produced using a Fourier spectral discretization, Figure 8b shows solutions produced using the finite difference discretization. The value of n_y for the finite difference method has been scaled by a factor of $2\pi/4$, so that the x-axes represent the same effective resolution.

5.3. Results: Stokes flow around a cylinder

We now compare the results from the Fourier discretization defined in Section 5.1 and the finite difference discretization defined in Section 5.2, across the $\text{IB}_{\sqrt{\cdot}}$, IBSE-1, and IBSE-2 methods. A convergence study showing the difference between successively refined solutions for u and p in $L^\infty(\Omega)$ is shown in Figure 8. The same trends in convergence as observed in Figure 4b are observed for the Fourier spectral discretization. For the finite difference discretization, the $L^\infty(\Omega)$ convergence is limited by the accuracy of the underlying discretization to second-order. Note, however, that the IBSE-1 and IBSE-2 methods still differ in the convergence of the pressure function p: pointwise convergence of the pressure function is first-order for the IBSE-1 method and second-order for the IBSE-2 method. The convergence rates for all elements of the stress tensor are the same as for the pressure.

The drag coefficient is a simple, commonly used benchmark for comparing solutions across methods, and is computed as

$$C_D = \frac{2F_D}{\overline{U}D}, \tag{45}$$

where $\overline{U}$ is the average velocity of the fluid across the inflow boundary, D is the diameter of the cylinder, and F_D is the drag force:

$$F_D = \int_\Gamma \sigma \cdot \mathbf{n} \cdot \hat{x}\, ds. \tag{46}$$

For IBSE-1 and IBSE-2, the elements of the stress tensor are captured accurately up to the boundary. To compute σ, we simply compute the appropriate derivatives of $\mathbf{u}$ (spectrally or with second-order finite differences *ignoring the boundary* Γ), and interpolate to the boundary by applying the operator S^*. For the spectral method, βx must be added to the pressure, as the constant body forcing $\beta\hat{x}$ that is applied to drive the flow acts as an effective pressure gradient. For the IB method, $\mathbf{u}$ is only continuous, and differentiation produces large errors at the interface. The drag force can, however, be found by integrating the singular forces supported along the boundary [10]:

$$F_D = \int_\Gamma \mathbf{G}(s) \cdot \hat{x}\, ds, \tag{47}$$

Method	Spectral				Finite Difference			
	n_y	C_D	% Error	Improvement	n_y	C_D	% Error	Improvement
$\text{IB}_{\sqrt{\cdot}}$		181.956	37.47	-		159.357	20.40	-
IBSE-1	64	129.564	2.11	18	32	124.161	6.19	3
IBSE-2		132.781	0.32	117		129.270	2.33	8
$\text{IB}_{\sqrt{\cdot}}$		141.505	6.91	-		137.860	4.16	-
IBSE-1	256	132.437	0.058	119	128	131.228	0.86	5
IBSE-2		132.361	≤ 0.004	≥ 1700		132.202	0.12	35
$\text{IB}_{\sqrt{\cdot}}$		134.495	1.61	-		133.657	0.98	-
IBSE-1	1024	132.425	0.049	33	512	132.142	0.16	6
IBSE-2		132.360	≤ 0.004	≥ 400		132.348	0.0091	108

Table 1: The drag coefficient C_D and % error, computed with respect to the reference value of 132.60 from [43]. Results are provided for the $\text{IB}_{\sqrt{\cdot}}$, IBSE-1, and IBSE-2 methods, for both the spectral and finite difference discretizations, and for a coarse, medium, and fine discretization. For the IBSE-1 and IBSE-2 methods, the improvement in the percent error, as compared to the percent error produced by the $\text{IB}_{\sqrt{\cdot}}$ method, is shown in the column labeled 'Improvement'. When $n_y = 64$ for the spectral discretization, $\Delta x = 2\pi/64 \approx 0.098$; when $n_y = 32$ for the finite difference discretization $\Delta x = 4/32 = 0.125$.

where $\mathbf{G}$ are the singular forces from Equation (17).

In Figure 9, we show a refinement study for the drag coefficient, as compared to the reference value of 132.36, from [43][7]. The IB method achieves first-order convergence of the drag coefficient in both methods; the IBSE-1 and IBSE-2 methods achieve first- and second-order convergence with the finite difference discretization and convergence that is at least first- and second-order with the Fourier spectral method. It is surprising that the drag coefficient for the IB method converges at all: none of the elements of the stress tensor converge pointwise up to the boundary. Nevertheless, the singular forces accurately represent the stress the fluid is exerting on the solid. Individual elements of the stress tensor, however, do not converge near to the boundary. For the IBSE-k method, the velocity field is $C^k(C)$, while the elements of the stress tensor are $C^{k-1}(C)$. For functions with $C^{k-1}(C)$ regularity, the operator S^* is accurate to $\mathcal{O}(\Delta x^k)$.

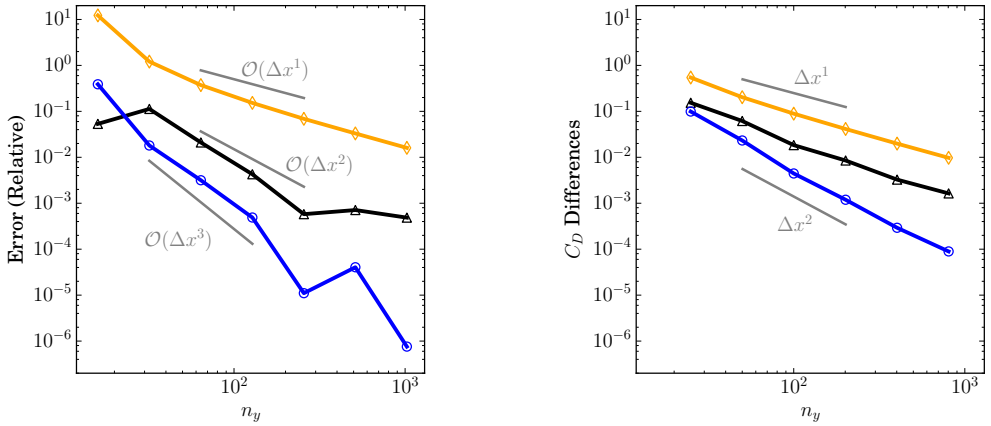
In Table 1, we show the drag coefficient and the percent error computed by the Immersed Boundary, IBSE-1, and IBSE-2 methods for both the spectral and finite difference discretizations over a range of values of n_y. The percent error is computed relative to the value of 132.36 from [43]. For the IBSE-1 and IBSE-2 methods, we also show the improvement in % error as compared to the $\text{IB}_{\sqrt{\cdot}}$ method. The IBSE method provides the greatest advantage when tight error tolerances are required. Using the finite difference discretization, to achieve a 1% error, the $\text{IB}_{\sqrt{\cdot}}$ method would need to use $n_y \approx 512$, while the IBSE-2 method requires only $n_y \approx 64$; however to achieve an error of 0.01%, the $\text{IB}_{\sqrt{\cdot}}$ method requires $n_y \approx 65536$, while IBSE-2 requires only $n_y \approx 512$.

5.4. A comparison of the numerical requirements

In Section 5.1, we discretized and solved a flow-around a cylinder problem in a confined channel using a Fourier spectral method to discretize the underlying differential operators. This provides a convenient and simple way to discretize the Stokes equations to high-order, but it comes with a significant drawback: the Dirichlet $\mathbf{u} = 0$ boundary condition along the geometrically simple channel walls must be enforced by adding a force distribution to the non-physical region where $y < -2\zeta$ or $y > 2\zeta$. This causes two significant inefficiencies:

1. Spatial degrees of freedom (for $\mathbf{u}$, p, $\boldsymbol{\xi}_{\mathbf{u}}$, ξ_p) must be solved for in the extension regions where $y < -2\zeta$ or $y > 2\zeta$.

[7] This geometry is typically used to study polymeric fluids. The most accurate results currently available are from a recent high-order hp spectral element method [39], but a result for the Newtonian case is not provided in that study. The value that we use for the Newtonian case comes from an older hp finite element method [43]. At low elasticity where both [43] and [39] report results, the drag coefficients reported agree to at least two decimal places.



(a) Spectral

(b) Finite difference

Figure 9: Refinement study showing the convergence of the drag coefficient to the value 132.36 from [43], produced by the IB$_{\sqrt{\cdot}}$(◊), IBSE-1 (△), and IBSE-2 (○) methods. The error is normalized by the reference value. The value of n_y for the finite difference method has been scaled by a factor of $2\pi/4$, so that the x-axes represent the same effective resolution.

2. Additional boundary nodes must be placed along the interior boundaries at $y = 2\zeta$ and $y = -2\zeta$ (corresponding to the channel walls).

The additional boundary nodes along the channel walls can become a significant problem for fine discretizations. Boundary nodes are placed a distance Δs apart, where Δs is set to be approximately twice the grid spacing: $\Delta s \approx 2\Delta x$. For this geometry, when n_y boundary points are used to discretize the spatial grid in the y direction, $3n_y$ boundary nodes are required for each of the channel walls.

In Table 2, we summarize the number of boundary nodes required for the computation, the size of the Schur-Complement that must be inverted for IBSE-1 and IBSE-2, and the time required to form the Schur complement as well as to compute the solution to one Stokes problem once the Schur complement has been formed. For the steady Stokes problem, the time to the form the Schur complement dominates the computation, the cost of a single Stokes solve is much smaller. For time dependent problems, the Schur-complement need only be formed once, facilitating efficient time-stepping.

For the finest discretization that we run for the spectral discretization ($n_y = 1024$), the IBSE-2 method requires a Schur-complement that is 53011×53011; the finite-difference scheme at this discretization requires a Schur-complement that is only 3856×3856. This can be a substantial issue. The matrix for the Fourier discretization requires approximately 2.9 billion elements to be stored; using double precision, this requires over 22 gigabytes of RAM. Although this is feasible in two dimensions, the discretization of long channel walls would be computationally prohibitive in three dimensions. For the finite difference discretization of the same size, only about 120 megabytes of memory are required.

Despite the disparity in memory requirements, solving the Stokes equations using a Fourier spectral discretization is very efficient, and the cost of each full solve is between 10 and 20 times faster using the spectral discretization. The time required to form the Schur complement is approximately equivalent: the much larger Schur-complement size is offset by the speed of the solve itself. This disparity is largest for the Stokes problem. For Navier-Stokes problems with moderate Reynolds numbers, the finite-difference solves are approximately 10 times faster due to the fact that fewer iterations of the preconditioned GMRES algorithm are required to reach the same tolerance. In this case the spectral solve is only moderately faster per timestep, and the finite-difference discretization enjoys a significant advantage in the time required to form the Schur-complement.

n_y	n_{wall}	n_{cylinder}	n_{bdy}	SC size		Solve Time		Formation Time	
				Spectral	FD	Spectral	FD	Spectral	FD
16	48	7	103	827	56	53	272	40000	15000
32	96	15	207	1659	120	58	568	100000	70000
64	192	30	414	3315	240	66	465	220000	110000
128	384	60	828	6627	480	56	630	370000	300000
256	768	120	1656	13251	960	43	640	570000	610000
512	1536	241	3313	26507	1928	38	722	1000000	1400000
1024	3072	482	6626	53011	3856	40	812	2100000	3100000
						(In units of time to compute an FFT)			

Table 2: A summary of the number of boundary nodes required to discretize the flow around the cylinder problem studied in Section 5, along with the associated size of the Schur-complement matrix defined in Equation (34) for both the Fourier spectral and finite-difference discretizations. The time required to solve one problem, as well as to form the entire Schur complement, normalized by the time required to execute one (real) FFT are also shown. The columns, from left to right, give n_y, the number of points used to discretize the computational domain C in the spanwise direction; n_{wall}, the number of boundary nodes used to discretize each wall; n_{cylinder}, the number of boundary nodes used to discretize the cylinder; n_{bdy}, the number of boundary nodes used to discretize all of the boundaries (for the spectral discretization), the size of the Schur complement (for both the spectral and finite-difference discretizations), and the (estimated) time to form the Schur-complement operator, again normalized by the time to execute one real FFT.

6. Navier-Stokes equations

Because the IBSE method enables the efficient solution of the Stokes problem in the domain Ω, unsplit time-stepping for the Navier-Stokes problem is simple when the non-linear terms are treated explicitly in time. We integrate the Navier-Stokes equation in time using a fourth-order implicit-explicit (IMEX) Backward Differentiation formula [44]:

$$\frac{25}{12}\mathbf{u}^{n+1} - 4\mathbf{u}^n + 3\mathbf{u}^{n-1} - \frac{4}{3}\mathbf{u}^{n-2} + \frac{1}{4}\mathbf{u}^{n-3} = \Delta t\left[\mathcal{I}(\mathbf{u}^{n+1}) + 4\mathcal{E}(\mathbf{u}^n) - 6\mathcal{E}(\mathbf{u}^{n-1}) + 4\mathcal{E}(\mathbf{u}^{n-2}) - \mathcal{E}(\mathbf{u}^{n-3})\right], \quad (48)$$

where $\mathcal{I}$ evaluates the *stiff* terms in $\mathbf{u}_t$ while $\mathcal{E}$ evaluates the *non-stiff* terms in $\mathbf{u}_t$. For this problem, $\mathcal{I}(\mathbf{u})$ is the Stokes operator and $\mathcal{E}(\mathbf{u}) = -\mathbf{u}\cdot\nabla\mathbf{u}$, discretized using second-order centered differences. The use of this type of timestepping eliminates errors due to a projection step and removes the need for artificial pressure boundary condtions, and is appropriate for low-Reynolds number problems and for problems where accuracy near to the boundary is critical.

To test the performance of the IBSE method on a time dependent problem, we set up and solve a flow around a cylinder problem in a standard benchmark flow geometry [45], shown in Figure 10. A variety of simple scalar benchmarks for this problem are available and are reported as ranges which ensure a percent error of at most 0.15% to 3%, depending on the benchmark [45]. To demonstrate the validity of the timestepping scheme, we first demonstrate the solver for a problem with an unsteady solution, at Reynolds number 100, and show quantitative agreement with time-dependent benchmarks. Unfortunately, very high-accuracy benchmarks are unavailable for this problem. To provide a better measure of the accuracy of the IBSE method, we solve a steady problem, with Reynolds number 20. For some subset of the benchmarks that we examine for this problem, results accurate to nearly machine precision are available from high-order spectral computations [46].

For both the unsteady and steady cases, the test problem is the confined flow around a cylindrical obstacle. The computational domain C is $[0, 2.05]\times[0, 0.41]$, the extension region $E = B_{0.05}(0.2, 0.2)$ is the cylinder of radius 0.05 centered at $(0.2, 0.2)$, and the physical domain $\Omega = C\setminus E$. We will denote the internal boundary $\partial B_{0.05}(0.2, 0.2)$ by Γ, while the boundary of the computational domain C will be denoted by ∂C. ∂C is divided into four parts: $\partial C = \cup\{\partial C_l, \partial C_r, \partial C_b, \partial C_t\}$, denoting the boundaries at $x = 0$, $x = 2.05$, $y = 0$, and $y = 0.41$, respectively. No-slip boundary conditions are applied on ∂C_b and ∂C_t, homogeneous Neumann boundary conditions are applied on ∂C_r, and the inflow condition $\mathbf{u} = (4sU_m y(0.41 - y)/0.41^2, 0)$

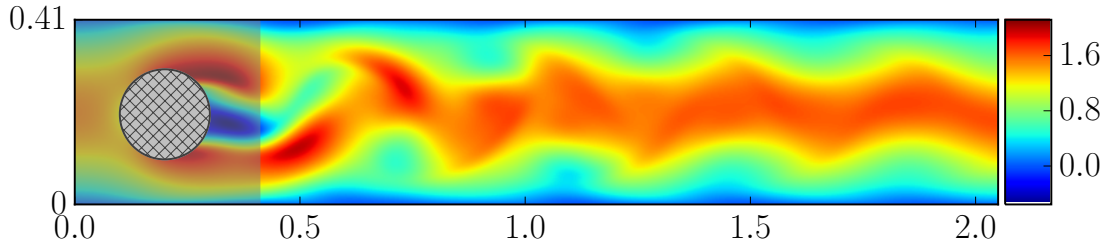


Figure 10: The solution u to the problem solved in Section 6.1, at $t = 12.0$, produced using a second-order finite difference discretization to the Navier-Stokes equations as the underlying PDE solver for the IBSE-2 method with $n_y = 512$. The extension region E is shown in white with hatching. The extension functions $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p are not solved for on the entirety of the computational domain, but rather in the much smaller region $C_E = [0.41, 0.41] \times [0.41, 0.41]$, localized around E, that is shaded in light gray.

is applied on ∂C_l, where U_m is the maximum value of the u velocity at inflow and will be chosen to set the Reynolds number. To be precise, we solve

$$\begin{aligned}
\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p &= 0 && \text{in } \Omega, && \text{(49a)}\\
\nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega, && \text{(49b)}\\
\mathbf{u} &= 0 && \text{on } \Gamma,\ \partial C_b, \text{ and } \partial C_t, && \text{(49c)}\\
u(y) &= 4U_m y(0.41 - y)/0.41^2 && \text{on } \partial C_l, && \text{(49d)}\\
v(y) &= 0 && \text{on } \partial C_l, && \text{(49e)}\\
\partial \mathbf{u}/\partial n &= 0 && \text{on } \partial C_r. && \text{(49f)}
\end{aligned}$$

For the IBSE method, C_E, the computational domain for computing the extension functions $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p is chosen to be $[0, 0.41] \times [0, 0.41]$. This geometry, along with the u value of the solution for the unsteady Reynolds number 100 case, produced by the IBSE-2 method with $n_y = 512$, is shown in Figure 10. We note that our geometry is slightly different than the benchmark geometry used in [45]. Their domain is taken to be a rectangle with width 2.2 and height 0.41, our rectangle is slightly less wide so that the rectangle has an aspect ratio of 5, allowing the grid spacing in the x direction and y direction to be equal.

In practice, timestepping involves the explicit computation of a forcing function:

$$\frac{25}{12}\mathbf{f}_{\mathbf{u}} = 4\mathbf{u}^n - 3\mathbf{u}^{n-1} + \frac{4}{3}\mathbf{u}^{n-2} - \frac{1}{4}\mathbf{u}^{n-3} + \Delta t \left[4\mathcal{E}(\mathbf{u}^n) - 6\mathcal{E}(\mathbf{u}^{n-1}) + 4\mathcal{E}(\mathbf{u}^{n-2}) - \mathcal{E}(\mathbf{u}^{n-3})\right], \qquad (50)$$

followed by solution of the equation

$$\begin{aligned}
\left(\mathbb{I} - \frac{12}{25}\nu \Delta t \Delta\right) \mathbf{u}^{n+1} + \nabla p^{n+1} &= \mathbf{f}_{\mathbf{u}} && \text{in } \Omega, && \text{(51a)}\\
\nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega, && \text{(51b)}\\
\mathbf{u}^{n+1} &= 0 && \text{on } \Gamma,\ \partial C_b, \text{ and } \partial C_t, && \text{(51c)}\\
u^{n+1}(y) &= 4U_m y(0.41 - y)/0.41^2 && \text{on } \partial C_l, && \text{(51d)}\\
v^{n+1}(y) &= 0 && \text{on } \partial C_l, && \text{(51e)}\\
\partial \mathbf{u}^{n+1}/\partial n &= 0 && \text{on } \partial C_r. && \text{(51f)}
\end{aligned}$$

6.1. Results: Unsteady flow (Reynolds number 100)

To achieve a Reynolds number of 100 for the unsteady case, U_m is set to 1.5 in Equation (51d). For this problem the Reynolds number is defined as $Re = \overline{U}D/\nu$, where $\overline{U}$ gives the average velocity across the inflow boundary. For the prescribed boundary condition, $\overline{U} = \frac{2}{3}U_m$. We integrate Equation (49) to $t = 12.0$ with

	C_D	C_L	Strouhal Number	ΔP
Reference Range	$[3.22, 3.24]$	$[0.99, 1.01]$	$[0.295, 0.305]$	$[2.46, 2.50]$
IBSE-2, $n_y = 128$	3.3123	0.9802	0.3005	2.4770
IBSE-2, $n_y = 512$	3.2338	0.9866	0.3018	2.4979

Table 3: The drag coefficient (C_D), lift coefficient (C_L), Strouhal number, and pressure difference computed by the IBSE-2 method for the unsteady Navier Stokes problem with Reynolds number 100. The results are shown for $n_y = 128$ and $n_y = 512$, and compared to the reference ranges provided in [45].

a timestep of $\Delta t = \Delta x/5$, for $n = 2^7$ and $n = 2^9$, using the IBSE-2 method when solving Equation (51). The timestep Δt need not be taken so small at this Reynolds number, but is chosen conservatively to ensure that error is spatially dominated. The u velocity at $t = 12.0$ is shown in Figure 10.

The benchmarks measured in [45] for this problem include the maximum drag coefficient (C_D), the maximum lift coefficient (C_L), the Strouhal number St, and the pressure difference (ΔP), measured at $t = t_0 + 1/(2f)$, where f is the frequency of separation and t_0 is the time at which the lift coefficient is maximized. The benchmark ranges, along with the values computed using the IBSE-2 method with $n_y = 128$ and $n_y = 512$ are shown in Table 3. For the IBSE-2 method with $n = 512$, the computed values of the benchmarks are consistent with the benchmark range, although the maximum value for the lift coefficient is slightly low. We note that method 9a from [45], whose values agree very well with our results and the high-order results from [46] for the steady Re = 20 case (see Section 6.2) also reports a lower maximum value of $C_L = 0.9862$ that is in strong agreement with the IBSE-2, $n_y = 512$ result for $C_L = 0.9866$.

6.2. Results: Steady flow (Reynolds number 20)

For a steady case, U_m is set to 0.3 [45], and ν is set to 0.001, yielding a Reynolds number $Re = \overline{U}D/\nu = 20$. We integrate Equation (49) to $t = 12.0$ with a timestep of $\Delta t = \Delta x/5$, for $n = 2^5$ to $n = 2^9$, using the IB method as well as IBSE-1 and IBSE-2 when solving Equation (51). As with the unsteady case, the timestep need not be taken so small but is chosen conservatively to ensure that error is spatially dominated. Convergence is assessed by comparing the ratio of the $L^\infty(\Omega)$ difference between solutions at successive levels of refinement (bilinear interpolation is used to transfer data between grids to do the refinement study). Results are shown in Figure 12. As expected, the IB method achieves first order convergence in $L^\infty(\Omega)$ for the velocity, but fails to converge pointwise for the pressure. Both the IBSE-1 and IBSE-2 methods converge at second-order for the velocity field. Although the IBSE-2 method produces $C^2(C)$ solutions, third order convergence is not possible, as the underlying finite-difference discretization is only accurate to second order. Despite asymptotically similar convergence, the actual error is lower for all discretizations. For the pressure, IBSE-1 converges at first order and IBSE-2 converges at second order.

To emphasize the improved convergence of the IBSE method for elements of the stress tensor, we examine the convergence of the pressure field p, interpolated to the boundary, in Figure 12b. Convergence is assessed

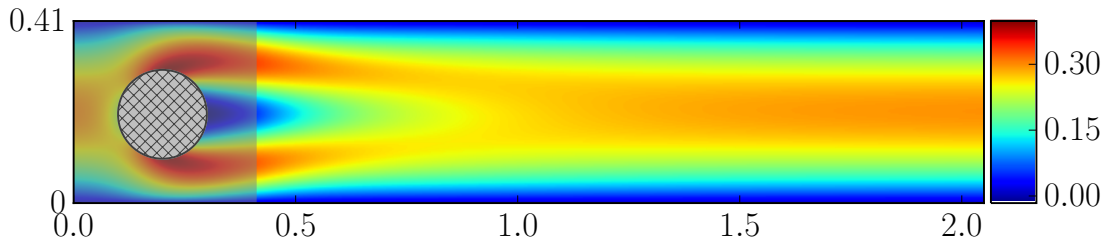
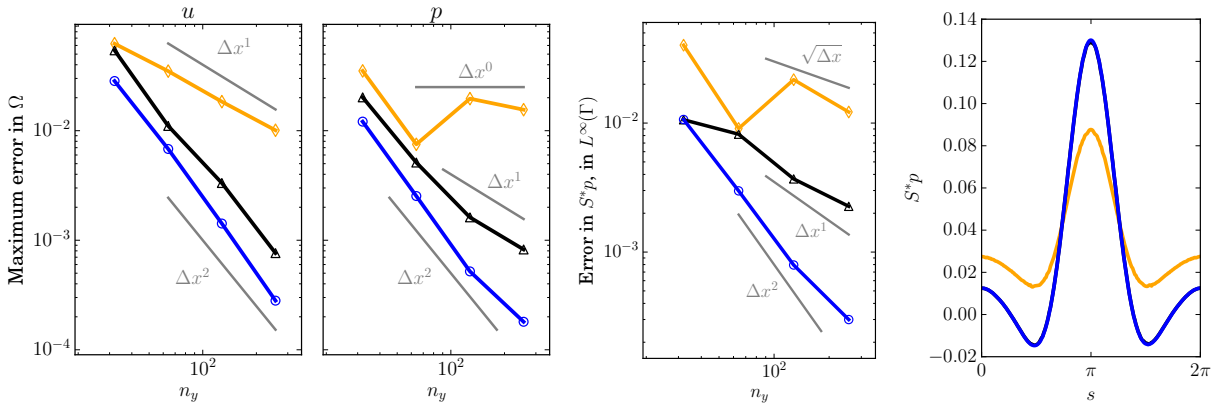


Figure 11: The solution u to the problem solved in Section 6.2, at steady state, produced using a second-order finite difference discretization to the Navier-Stokes equations as the underlying PDE solver for the IBSE-2 method with $n_y = 512$. The extension region E is denoted with crosshatches. The extension functions $\boldsymbol{\xi}_{\mathbf{u}}$ and ξ_p are not solved for on the entirety of the computational domain, but rather in the much smaller region $C_E = [0.41, 0.41] \times [0.41, 0.41]$, localized around E, that is shaded in light gray.



(a) Refinement study for u, p in $L^\infty(\Omega)$ (b) $L^\infty(\Gamma)$ difference in S^*p (c) $S^*p(s)$

Figure 12: Refinement study showing the convergence of differences between successively refined solutions for u and p in $L^\infty(\Omega)$ (Figure 12a), and for S^*p in $L^\infty(\Gamma)$ (Figure 12b), to the steady Navier-Stokes problem defined in Equation (49) at steady state ($t = 12.0$), produced using the IB$_{\sqrt{\cdot}}$(◊), IBSE-1 (△), and IBSE-2 (○) methods. Figure 12c shows the pressure function interpolated to the boundary, although this converges to something, as shown in Figure 12b, it is not converging to the correct value. Note that the lines for IBSE-1 and IBSE-2 visually coincide.

by comparing the ratio of the $L^\infty(\Gamma)$ difference between S^*p at different resolutions. From examining Figure 12b, it appears that the IB method converges, albeit slowly. However, looking at Figure 12c, we see that although the IB method is converging to *something* on the boundary, that value is not consistent with the value determined by the IBSE method. For the IBSE-1 and IBSE-2 methods, S^*p converges at first- and second-order, respectively. We note that the benchmark pressure difference, ΔP, for the IBSE-1 and IBSE-2 methods is consistent with the ranges reported in [45] (see Table 4), giving us good reason to believe that the pressure values (up to a constant) on the cylinder shown in Figure 12c are accurate.

For this problem the availability of high-accuracy values [46] for the drag coefficient (C_D), the lift coefficient (C_L) and the pressure difference (ΔP) allows for precise benchmarking. We will also compute the length of the recirculation zone (L_a), for which only a benchmark range [45] is available. For the IBSE method, the drag and lift coefficients are computed by directly computing the stress, interpolating to the boundary, and computing the appropriate integral over Γ. For the IB method, this provides an inconsistent estimate, the appropriate integral over the singular forces is computed instead [10]. No pressure difference (ΔP) is reported for the IB method, as a consistent estimate of the pressure on the boundary cannot be obtained. This comparison is shown in Table 4. The IB method does surprisingly well for all benchmarks: despite the fact that components of the stress tensor cannot be directly evaluated on the boundary, the singular forces $\mathbf{G}$ provide an alternative measure that is convergent. For C_D, C_L, and ΔP, we expect the IBSE-1 method to provide first-order estimates due to the fact that the stress σ is only continuous across the boundary Γ. The estimates for the benchmarks have approximately the same accuracy as the IB estimates. For the IBSE-2 method, the stress is differentiable across the boundary, and we expect these benchmarks to be second-order accurate. Unsurprisingly this method provides much more accurate values. For $n = 512$, the IBSE-2 method yields a % error in the drag coefficient that is nearly two orders of magnitude smaller than the % error provided by the IB method. We again emphasize that with the IB method, consistent estimates of p cannot be obtained at the boundary, and so we do not attempt to calculate a pressure difference ΔP for these methods.

7. Discussion

In [27], we introduced the IBSE method for solving elliptic and parabolic PDE on general smooth domains to an arbitrarily high order of accuracy using simple Fourier spectral methods. In this paper, we extend the

Benchmark		C_D		C_L		L_a	ΔP	
Range		$[5.57, 5.59]$		$[0.0104, 0.0110]$		$[0.0842, 0.0852]$	$[0.1172, 0.1176]$	
Value		5.57953523384		0.010618948146		-	0.11752016697	
		value	$\%_{\text{err}}$	value	$\%_{\text{err}}$	value	value	$\%_{\text{err}}$
$n = 128$	$\text{IB}_{\sqrt{\cdot}}$	5.76813	3.38003	0.01306	22.9463	0.08803	-	-
	IBSE-1	5.52533	0.97142	0.00405	61.8221	0.08514	0.11625	1.08426
	IBSE-2	5.58697	0.13323	0.00875	17.5530	0.08481	0.11714	0.32420
$n = 512$	$\text{IB}_{\sqrt{\cdot}}$	5.62468	0.80919	0.01081	1.81973	0.08567	-	-
	IBSE-1	5.55464	0.44618	0.01228	15.6430	0.08465	0.11677	0.63829
	IBSE-2	5.57883	0.01271	0.01069	0.71540	0.08464	0.11759	0.06075

Table 4: The drag coefficient (C_D), lift coefficient (C_L), length of the recirculation zone (L_a), and the pressure difference (ΔP), computed by the $\text{IB}_{\sqrt{\cdot}}$, IBSE-1, and IBSE-2 methods, compared to the benchmark ranges computed in [45] (labeled as "Benchmark Range"), as well as the results from the high-order spectral computations in [46] (labeled as "Benchmark Value"). The percent errors reported are computed in reference to the benchmark value.

methodology to fluid problems with divergence constraints, including the incompressible Stokes and Navier Stokes equations. We decouple the extension procedure from the underlying PDE solver, and demonstrate accurate solutions with an underlying Fourier spectral solver as well as a second-order finite difference method. For the velocity field, we demonstrate up to third-order pointwise convergence, in contrast to the first-order pointwise convergence of the IB method. In addition, we show up to second-order pointwise convergence of the pressure function and all elements of the fluid stress tensor. In the IB method, these quantities do not converge pointwise.

In the IBSE method, accuracy is obtained by forcing the solution to be globally smooth on the entire computational domain. This is accomplished by solving for the smooth extension of the *unknown* solution, allowing for application of the IBSE method to the steady Stokes equations. Remarkably, the fully coupled problem for the solution to the incompressible Stokes equations (combined with the equations that define the smooth extensions to the velocity and pressure fields) can be effectively reduced to a small, dense system of equations, and the cost to invert this system can be minimized by precomputing and storing its LU-factorization. The IBSE method thus provides an efficient algorithm for implicit or implicit-explicit (IMEX) timestepping on stationary domains. The method requires minimal geometric information regarding the boundary: only its position, normals, and an indicator variable denoting whether points in the computational grid are inside of the physical domain Ω or not, enabling simple and robust code. The method is flexible enough to allow high-order discretization in space and time for a wide range of nonlinear PDE using straightforward implicit-explicit timestepping schemes.

The IBSE method decouples the discretization of the extension problem from the discretization of the underlying PDE. By itself, this allows some gains in efficiency when the extension region is small compared to the physical domain Ω, as is often the case when studying the flow around objects. More importantly, however, the methodology can be easily coupled to existing codes. This enables the simple and accurate inclusion of embedded boundaries into ongoing research without significant redevelopment of code. Additionally, it allows for the ability to build on top of validated and optimized code for the solution of PDE, including those that that use adaptive mesh refinement, without sacrificing accuracy near to the embedded boundary.

The IBSE method shares some similarities with Fourier Continuation (FC) [23–25] and Active Penalty (AP) [26] methods. The FC method uses Fourier methods to obtain a high-order discretization, and the idea that smooth, non-periodic functions can be turned into smooth and periodic functions by extending them into a larger domain in some appropriate way. There are two key differences between the FC method and the IBSE method. The FC method (*i*) uses dimensional splitting to reduce the problem to a set of one-dimensional problems and (*ii*) relies on data from the previous timestep in order to generate the Fourier continuations that allow for their high-order spatial accuracy. This forces the FC method to use an Alternating-Direction Implicit scheme to take large stable timesteps when solving parabolic equations, complicating the implementation of high-order timestepping. In addition, the FC method requires the use

of an iterative solver for computing solutions to the Poisson equation, complicating an efficient discretization of the *incompressible* Navier-Stokes equations, although the FC method has been used to solve the compressible Navier-Stokes equations to high order [23]. In contrast to the conditioning issues faced by the IBSE method, the one-dimensional nature of the Fourier-continuation problem allows the FC method to use high-precision arithmetic in certain precomputations to effectively eliminate the conditioning problem inherent in constructing smooth extensions [47]. This enables the FC method to obtain stable methods that converge to higher order than can be achieved by the IBSE method in double-precision arithmetic.

The AP method, like the FC method, relies on data from previous timesteps in the way that it imposes smoothness on the solution of the PDE. A large drag force is applied that penalizes deviations of the solution in the extension region from the smooth extension of the solution at the previous timestep. This dependence on data from previous timesteps forces the AP method to use *explicit* timestepping when solving parabolic equations. The AP algorithm for the Navier-Stokes equations uses a projection method, requires explicit time-stepping, and is only able to achieve second- and first-order convergence for the velocity and pressure fields, respectively.

In contrast, the IBSE method directly solves the steady Stokes problem, smoothly extending both the unknown velocity and pressure fields. The extension equations are coupled directly to the unknown solutions through the unsplit Stokes problem, eliminating the errors inherent in projection methods at low Reynolds number. The price that we pay for extending the unknown solution is that the high-efficiency that we obtain is only currently achievable on stationary domains. This is because the inversion method used for the IBSE method relies on an expensive precomputation that depends on the physical domain and the discretization. However, there is no fundamental obstacle to the application of the IBSE method to problems with moving domains. Instead, the challenge is to find a robust and efficient method to invert the IBSE system in Equation (25) that does not require substantial precomputation. Recent progress has been made for preconditioning similar systems of equations for the simulation of rigid-body motion in an Immersed Boundary framework [4]. The integration of these ideas into the IBSE method to allow for simulation of moving boundary problems will be an area of active future research.

We have only implemented two-dimensional examples in this paper. The IBSE method extends to three dimensions without any changes, although there is one minor difficulty that must be resolved: the production of an accurate enough quadrature for the discretization of the boundary integrals in the spread operators S and T_k. For two-dimensional problems, this comes nearly for free since the simple quadrature rule given in Section 3.1 is spectrally accurate for closed one-dimensional curves. High-order quadrature rules for two-dimensional surfaces are more complicated but well-developed, and high-order surface representation has been incorporated into the Immersed Boundary method [48, 49].

Acknowledgements

This work was supported in part by the National Science Foundation under Grant DMS-1160438.

[1] Charles S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002. ISSN 0962-4929. doi: 10.1017/S0962492902000077.

[2] Rajat Mittal and Gianluca Iaccarino. Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, 2005. ISSN 0066-4189. doi: 10.1146/annurev.fluid.37.061903.175743.

[3] Sarah D Olson and Anita T Layton. Simulating biofluid-structure interactions with an immersed boundary framework–a review. *Biological Fluid Dynamics: Modeling, Computations, and Applications*, 628:1, 2014.

[4] 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, 2016.

[5] Kunihiko Taira and Tim Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007. ISSN 00219991. doi: 10.1016/j.jcp.2007.03.005.

[6] Joseph M Teran and Charles S Peskin. Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain. *SIAM Journal on Scientific Computing*, 31(5):3404–3416, 2009.

[7] Zhilin Li and Kazufumi Ito. *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*, volume 33. Siam, 2006.

[8] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152(2):457–492, 1999. ISSN 00219991. doi: 10.1006/jcph.1999.6236. URL http://www.sciencedirect.com/science/article/pii/S0021999199962368.

[9] Philippe Angot, Charles-Henri Bruneau, and Pierre Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4):497–520, 1999. ISSN 0029-599X. doi: 10.1007/s002110050401.

[10] Ming-Chih Lai and Charles S. Peskin. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. *Journal of Computational Physics*, 160(2):705–719, 2000. ISSN 00219991. doi: 10.1006/jcph.2000.6483. URL http://linkinghub.elsevier.com/retrieve/pii/S0021999100964830.

[11] Andreas Mark and Berend G M van Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227(13):6660–6680, 2008. ISSN 00219991. doi: 10.1016/j.jcp.2008.03.031.

[12] Mark N. Linnick and Hermann F. Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, 204(1):157–192, 2005. ISSN 00219991. doi: 10.1016/j.jcp.2004.09.017.

[13] Jian Kang Liu and Zhou Shun Zheng. Efficient high-order immersed interface methods for heat equations with interfaces. *Applied Mathematics and Mechanics*, 35(51174236):1189–1202, 2014. ISSN 02534827. doi: 10.1007/s10483-014-1851-6.

[14] Sheng Xu and Z. Jane Wang. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics*, 216(2):454–493, 2006. ISSN 00219991. doi: 10.1016/j.jcp.2005.12.016.

[15] Xiaolin Zhong. A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity. *Journal of Computational Physics*, 225(1):1066–1099, 2007. ISSN 00219991. doi: 10.1016/j.jcp.2007.01.017.

[16] S Yu, Y Zhou, and G Wei. Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces. *Journal of Computational Physics*, 224(2):729–756, 2007. ISSN 00219991. doi: 10.1016/j.jcp.2006.10.030. URL http://dx.doi.org/10.1016/j.jcp.2006.10.030.

[17] Y. C. Zhou, Shan Zhao, Michael Feig, and G. W. Wei. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics*, 213(1):1–30, 2006. ISSN 00219991. doi: 10.1016/j.jcp.2005.07.022.

[18] Frédéric Gibou and Ronald Fedkiw. A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *Journal of Computational Physics*, 202(2):577–601, 2005. ISSN 00219991. doi: 10.1016/j.jcp.2004.07.018.

[19] John P. Boyd. Fourier embedded domain methods: extending a function defined on an irregular region to a rectangle so that the extension is spatially periodic and C^∞. *Applied Mathematics and Computation*, 161(2):591–597, 2005. ISSN 00963003.

[20] Alfonso Bueno-Orovio. Fourier embedded domain methods: Periodic and c^∞ extension of a function defined on an irregular region to a rectangle via convolution with gaussian kernels. *Applied Mathematics and Computation*, 183(2):813–818, 2006. ISSN 00963003. doi: 10.1016/j.amc.2006.06.029.

[21] S. H. Lui. Spectral domain embedding for elliptic PDEs in complex domains. *Journal of Computational and Applied Mathematics*, 225(2):541–557, 2009. ISSN 03770427. doi: 10.1016/j.cam.2008.08.034. URL http://dx.doi.org/10.1016/j.cam.2008.08.034.

[22] Feriedoun Sabetghadam, Shervin Sharafatmandjoor, and Farhang Norouzi. Fourier spectral embedded boundary solution of the Poisson's and Laplace equations with Dirichlet boundary conditions. *Journal of Computational Physics*, 228(1):55–74, 2009. ISSN 00219991. doi: 10.1016/j.jcp.2008.08.018. URL http://dx.doi.org/10.1016/j.jcp.2008.08.018.

[23] Nathan Albin and Oscar P. Bruno. A spectral FC Solver for the compressible Navier-Stokes equations in general domains I: Explicit time-stepping. *Journal of Computational Physics*, 230(16):6248–6270, jul 2011. ISSN 00219991. doi: 10.1016/j.jcp.2011.04.023. URL http://linkinghub.elsevier.com/retrieve/pii/S0021999111002695.

[24] Mark Lyon and Oscar P. Bruno. High-order unconditionally stable FC-AD solvers for general smooth domains I. Basic Elements. *Journal of Computational Physics*, 229(9):3358–3381, 2010. ISSN 00219991. doi: 10.1016/j.jcp.2010.01.006. URL http://dx.doi.org/10.1016/j.jcp.2009.11.020.

[25] Mark Lyon and Oscar P. Bruno. High-order unconditionally stable FC-AD solvers for general smooth domains II. Elliptic, parabolic and hyperbolic PDEs; theoretical considerations. *Journal of Computational Physics*, 229(9):3358–3381, 2010. ISSN 00219991. doi: 10.1016/j.jcp.2010.01.006. URL http://dx.doi.org/10.1016/j.jcp.2010.01.006.

[26] David Shirokoff and J-C Nave. A Sharp-Interface Active Penalty Method for the Incompressible Navier-Stokes Equations.

Journal of Scientific Computing, 62(1):53—-77, 2015. URL http://arxiv.org/abs/1303.5681.

[27] David B. Stein, Robert D. Guy, and Becca Thomases. Immersed Boundary Smooth Extension: A high-order method for solving PDE on arbitrary smooth domains using Fourier spectral methods. *Journal of Computational Physics*, 304: 252–274, 2015. ISSN 10902716. doi: 10.1016/j.jcp.2015.10.023. URL http://arxiv.org/abs/1506.07561.

[28] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

[29] Mingchao Cai, Andy Nonaka, John B Bell, Boyce E Griffith, and Aleksandar Donev. Efficient variable-coefficient finite-volume stokes solvers. *Communications in Computational Physics*, 16(05):1263–1297, 2014.

[30] HC Brinkman. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Applied Scientific Research*, 1(1):27–34, 1949.

[31] Robert A Adams and John JF Fournier. *Sobolev spaces*, volume 140. Academic press, 2003.

[32] G. Akiki and S. Balachandar. Immersed boundary method with non-uniform distribution of Lagrangian markers for a non-uniform Eulerian mesh. *Journal of Computational Physics*, 307(November):34–59, 2016. ISSN 00219991. doi: 10.1016/j.jcp.2015.11.019. URL http://linkinghub.elsevier.com/retrieve/pii/S0021999115007597.

[33] David B Stein. *The Immersed Boundary Smooth Extension (IBSE) Method: A Flexible and Accurate Fictitious Domain Method, and Applications to the Study of Polymeric Flow in Complex Geometries*. PhD thesis, 2016.

[34] Fritz John. Partial differential equations, volume 1 of applied mathematical sciences, 1982.

[35] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback).

[36] Yoichiro Mori. Convergence proof of the velocity field for a stokes flow immersed boundary method. *Communications on Pure and Applied Mathematics*, 61(9):1213–1263, 2008. ISSN 00103640. doi: 10.1002/cpa.20233.

[37] Hua-shu Dou and Nhan Phan-thien. The flow of an Oldroyd-B fluid past a cylinder in a channel : adaptive viscosity vorticity (DAVSS- 3) formulation. 87(1999), 2006.

[38] M.a. Alves, F.T. Pinho, and P.J. Oliveira. The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods. *Journal of Non-Newtonian Fluid Mechanics*, 97(2-3):207–232, feb 2001. ISSN 03770257. doi: 10.1016/S0377-0257(00)00198-1. URL http://linkinghub.elsevier.com/retrieve/pii/S0377025700001981.

[39] S. Claus and T.N. Phillips. Viscoelastic flow around a confined cylinder using spectral/hp element methods. *Journal of Non-Newtonian Fluid Mechanics*, 200:131–146, oct 2013. ISSN 03770257. doi: 10.1016/j.jnnfm.2013.03.004. URL http://linkinghub.elsevier.com/retrieve/pii/S0377025713000785.

[40] J Eddie Welch, Francis Harvey Harlow, John P Shannon, and Bart J Daly. The mac method-a computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces. Technical report, Los Alamos Scientific Lab., Univ. of California, N. Mex., 1965.

[41] S McKee, MF Tomé, VG Ferreira, JA Cuminato, A Castelo, FS Sousa, and N Mangiavacchi. The MAC method. *Computers & Fluids*, 37(8):907–930, 2008.

[42] M. Cai, A. J. Nonaka, J. B. Bell, B. E. Griffith, and A. Donev. Efficient Variable-Coefficient Finite-Volume Stokes Solvers. pages 1–25, 2013. URL http://arxiv.org/abs/1308.4605.

[43] Yurun Fan, RI Tanner, and N Phan-Thien. Galerkin/least-square finite-element methods for steady viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, 84(2):233–256, 1999.

[44] Willem Hundsdorfer and Steven J. Ruuth. IMEX extensions of linear multistep methods with general monotonicity and boundedness properties. *Journal of Computational Physics*, 225(2):2016–2042, 2007. ISSN 00219991. doi: 10.1016/j.jcp.2007.03.003.

[45] M. Schäfer and S. Turek. Benchmark Computations of Laminar Flow Around a Cylinder. *Flow Simulation with High-Performance Computers II, Volume 52 of Notes on Numerical Fluid Mechanics, Vieweg*, 52:547 – 566, 1996. ISSN 0001-1452. doi: 10.1007/978-3-322-89849-4_39.

[46] Guido Nabh. *On high order methods for the stationary incompressible Navier-Stokes equations*. Interdisziplinäres Zentrum für Wiss. Rechnen der Univ. Heidelberg, 1998.

[47] Rodrigo B. Platte, Lloyd N. Trefethen, and Arno B. J. Kuijlaars. Impossibility of Fast Stable Approximation of Analytic Functions from Equispaced Samples. *SIAM Review*, 53(2):308–318, 2011. ISSN 0036-1445. doi: 10.1137/090774707.

[48] Boyce E Griffith and Xiaoyu Luo. Hybrid finite difference/finite element version of the immersed boundary method. *Submitted in revised form*, 2012.

[49] Varun Shankar, Grady B. Wright, Aaron L. Fogelson, and R. M. Kirby. A Study of Different Modeling Choices For Simulating Platelets Within the Immersed Boundary Method. pages 1–33, oct 2012. URL http://arxiv.org/abs/1210.1885v1.