The Immersed Boundary Double Layer (IBDL) Method

By

BRITTANY JAE LEATHERS
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

APPLIED MATHEMATICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

—————————————————————

Robert Guy, Chair

—————————————————————

Becca Thomases

—————————————————————

Gregory Miller

Committee in Charge

2022

i

To Colin

Abstract

The Immersed Boundary (IB) method of Peskin (J. Comput. Phys., 1977) is useful for problems that involve fluid-structure interactions or complex geometries. By making use of a regular Cartesian grid that is independent of the geometry, the IB framework yields a robust numerical scheme that can efficiently handle immersed deformable structures. Additionally, the IB method has been adapted to problems with prescribed motion and other PDEs with given boundary data. IB methods for these problems traditionally involve penalty forces which only approximately satisfy boundary conditions, or they are formulated as constraint problems. In the latter approach, one must find the unknown forces by solving an equation that corresponds to a poorly conditioned first-kind integral equation. This operation can therefore require a large number of iterations of a Krylov method, and since a time-dependent problem requires this solve at each step in time, this method can be prohibitively inefficient without preconditioning. In this dissertation, we introduce a new, well-conditioned IB formulation for boundary value problems, which we call the Immersed Boundary Double Layer (IBDL) method. We formulate the method for Poisson, Helmholtz, Brinkman, Stokes, and Navier-Stokes equations and demonstrate its efficiency over the original constraint method. In this double layer formulation, the equation for the unknown boundary distribution corresponds to a well-conditioned second-kind integral equation that can be solved efficiently with a small number of iterations of a Krylov method without preconditioning. Furthermore, the iteration count is independent of both the mesh size and spacing of the immersed boundary points. The method converges away from the boundary, and when combined with a local interpolation, it converges in the entire PDE domain. Additionally, while the original constraint method applies only to Dirichlet problems, the IBDL formulation can also be used for Neumann boundary conditions.

Acknowledgments

There are many people to whom I owe thanks, but I would like to thank two people in particular.

Thank you to my advisor, Bob Guy. You have given me a tremendous amount of your time, support, and guidance. You have pushed me to become a better mathematician and researcher. You have shown me patience and kindness when I needed it, and you have taught me what a difference a great mentor can make.

And thank you to my partner, Joshua Parker. Thank you for letting me run ideas by you, for cooking meals when research took up all my time, and for being there when I needed you. You have supported me in more ways than I can count. You have also introduced me to new ways of thinking and helped me grow as a person. Finally, you have believed in me even when my confidence has faltered, and for this I am exceedingly grateful.

Contents

viii

List of Figures

xi

List of Tables

Chapter 1

Introduction

The Immersed Boundary (IB) method [**45**, **47**] is a valuable numerical tool for fluid-structure inter-actions. It was initially developed by Peskin for problems involving elastic, deformable structures, such as those involved in cardiac dynamics [**28**, **41**, **42**, **44**, **46**]. However, the robustness and simplicity of the IB method has led to its use in many different applications (see [**16**, **19**, **56**], for just a few). Additionally, since its creation, there have been many developments and variations in the IB method. For instance, the IB framework has been altered to incorporate porous boundaries [**36**, **55**] and model elastic rods with a curvature or twist [**39**, **43**]. It has also been adapted to the flow of non-Newtonian fluids [**9**, **57**] and coupled with internal force mechanisms to model swimming organisms [**14**, **15**]. In recent years, work has been done to apply the Immersed Boundary framework to problems involving prescribed boundary values [**8**, **35**, **58**, **59**, **61**, **63**], which is the category of problems that this dissertation addresses.

The robustness of the IB method comes from its use of two coordinate systems: a Lagrangian system that moves with the structure and a fixed Eulerian system on which the fluid equations are solved. It uses convolutions with discrete delta functions to link these systems together and map forces from the structure to the grid. One can then solve the PDE on a regular Cartesian mesh. This elimination of the physical boundary makes it possible to use a PDE solver that is efficient and independent of the geometry of the structure.

As stated, this dissertation will focus on problems involving prescribed boundary values. In the field of fluid dynamics, this includes the motion of rigid bodies and fluid flow through domains with stationary boundaries or boundaries with prescribed motion. Since the IB method bypasses the need for a conforming mesh, it also has obvious advantages in the broader case of solving PDEs on complex domains.

When the IB method is applied to deformable structures, the boundary force density is found using a constitutive law from the structure characteristics [47], but in the case of rigid bodies, one needs a different way to find or interpret the boundary force. There have been several routes taken to use the IB framework in these situations. One method is to consider the Lagrangian points to be tethered to specified locations by springs. The boundary force is then interpreted as the spring restoring force, penalizing deviations from the prescribed boundary position [7, 20, 38, 60]. By using the IB framework, such penalty methods can be efficiently implemented, but they use parameters, such as spring constants, to approximate a rigid limit, and the required magnitude of the appropriate parameter leads to numerically stiff equations that necessitate very small time steps.

Another approach for applying the IB method to a rigid body problem consists of viewing the boundary force density as a Lagrange multiplier, used to enforce the no-slip boundary condition [59]. Chapter 2 gives a description of this method, and we refer to it as the Immersed Boundary Single Layer (IBSL) method. In this constraint method, the velocity and force are both unknowns in an algebraic system. One way to solve this system is to invert the Schur complement to solve first for the Lagrange multiplier force and then for the velocity. The main disadvantage of such a method is that this operator suffers from poor conditioning [35]. Furthermore, the conditioning of the discrete problem worsens when the Cartesian grid is refined or when the Lagrangian point spacing is refined relative to the grid. To avoid solving the fully discretized constraint problem, most numerical methods using this constraint approach rely on some form of time step splitting [8, 58, 59, 61, 63]. Taira and Colonius [59], for example, solve a simpler unconstrained system for an intermediate velocity, use this velocity to find pressure and the unknown boundary force, and then complete a projection step to remove non-divergence-free and slip components of the velocity. However, like the penalty methods, these fractional step methods only satisfy the constraint equations *approximately*, which can result in fluid penetration into a rigid body. Additionally, such a method cannot be used for steady Stokes or other time-independent PDE.

However, if instead of a time-splitting scheme, one inverts the Schur complement with a Krylov method, the number of iterations required could be very large due to its poor conditioning. The computational cost then becomes prohibitive for a time-dependent problem, when this solve would

be required at each time step. One way around this is to design a preconditioner [12,30,54], such as the physics-based approximation of the Schur complement constructed by Kallemov et al. [35]. A preconditioner can allow for a more efficient solution to the constraint problem, but developing and implementing one is a nontrivial undertaking, and such a preconditioner can involve computing the inverse of a dense matrix, which can be computationally expensive when the number of boundary points is large. Another limitation is that in order to control the conditioning, there is often a requirement that the boundary points be spaced to about twice the grid spacing, which, depending on the application and the Eulerian-Lagrangian coupling scheme, may result in decreased accuracy of the solution [18,27,35].

In this dissertation, we present a reformulated Immersed Boundary method for prescribed boundary values. Like the Immersed Boundary Single Layer method, our method enforces the boundary conditions *exactly*, but the resulting linear system is very well-conditioned. We can therefore solve it with an unpreconditioned Krylov method with very few iterations. This conditioning does not worsen as we refine the mesh, nor as we tighten the spacing of our boundary points relative to the grid. We are therefore able to avoid the need for a preconditioner altogether.

We formulate this new IB method by utilizing a connection between the IBSL method and a single layer boundary integral equation. Boundary integral methods rely on reformulating a boundary value problem as an integral equation with an unknown density on the boundary. These methods are particularly useful for linear, elliptic, and homogeneous PDE [51]. One advantage of these methods, which we exploit, is that there are well-conditioned integral representations available. For example, for the Helmholtz or Poisson equation, starting with a broad integral representation, one can derive single and double layer integral representations, the second of which produces an operator with much better conditioning. There are, however, some disadvantages to boundary integral methods. Firstly, once the boundary density is found, it is expensive to evaluate the solution on an entire grid, whereas by using efficient solvers, the IB method can do this quickly. Additionally, in order to directly implement integral methods, one needs an analytical Green's function for each specific PDE and problem domain, making nontrivial exterior boundary conditions complicated to implement. The IB method, on the other hand, does not require a Green's function and can be used on more general domains. The method we present in this paper

maintains the flexibility of the IB method while capturing the better conditioning of a double layer integral equation.

The connection between the IBSL method and a regularized single layer integral equation has been identified in a few recent works. Usabiaga et al. [62] illustrate this connection and discuss that their rigid multiblob method can be seen as a technique for solving a regularized first-kind integral equation for Stokes flow. Eldredge [17] made a more general connection between IB methods and boundary integral equations by extending the form of a PDE to govern a variable that defines a different function for each side of an immersed boundary. The resulting PDE contains jumps in field quantities across the boundary that correspond to the strengths of single and double layer potentials. This general connection allows for solutions on either side of the boundary. However, in this work, we look specifically at the IB method for prescribed boundary value problems for which the PDE domain exists on only one side of the boundary, and we create an IB formulation that corresponds to the use of a double layer potential with an unknown strength in order to take advantage of the better conditioning.

In this dissertation, we present the new *Immersed Boundary Double Layer (IBDL) method*, which is able to achieve the same order of accuracy as the original IBSL method away from the boundary, while only requiring a small number of iterations of a Krylov method. We derive, implement, and analyze this method for the 2-D Helmholtz, Poisson, Brinkman, Stokes, and Navier-Stokes equations with Dirichlet boundary conditions. While we focus on two dimensions in this dissertation, the method is not restricted to this case. Additionally, while the IBSL method applies only to Dirichlet problems, our new method can also be used for Neumann boundary conditions.

This dissertation is organized as follows. The first three chapters provide introductory and background material. In Chapter 2, we introduce the Immersed Boundary Single Layer method for solving PDEs with Dirichlet boundary conditions. In Chapter 3, we provide an introduction to Green's functions and boundary integral equations for the relevant PDEs.

In Chapter 4, we explicitly connect the IBSL method for Poisson and Stokes equations to corresponding single layer integral equations. In Chapter 5, we present the form of the Immersed Boundary Double Layer method as it applies to Poisson and Helmholtz equations. We then explicitly demonstrate that the method corresponds to a regularized double layer integral equation.

We then discuss a linear interpolation that we use in order to obtain convergence in the solution near the immersed boundary. Additionally, we form and analyze the IBDL method for Neumann boundary conditions. Lastly, the method is tested and analyzed on several scalar PDEs on interior and exterior domains. We demonstrate that the method achieves the same first-order accuracy as the IBSL method, while requiring far fewer iterations of a Krylov method.

The IBSL method has a very direct generalization from scalar PDEs to Stokes equation, in which fluid velocity and pressure are coupled. However, matching the form of the double layer integral representation for Stokes equation requires a non-trivial adaptation of the IBDL method. Chapter 6 presents the form of the IBDL method in this case and demonstrates its connection to a regularized double layer integral equation. We again test and analyze several PDEs and domains, including Stokes flow past a periodic array of cylinders, for which we compare drag forces against several numerical methods and asymptotic approximations. We additionally implement the IBDL method in the time-dependent Navier-Stokes equation and again compare drag forces with other numerical methods. Finally, in Chapter 7, we discuss the benefits and drawbacks of the IBDL method. We also discuss several directions that our future work will take in order to improve and generalize the IBDL method.

Chapter 2

The Immersed Boundary method for boundary value problems

In this chapter, we give an introduction to the Immersed Boundary (IB) method for solving elliptic Dirichlet boundary value problems. For a scalar function u and an elliptic operator $\mathcal{L}$, such a problem has the form

$$\mathcal{L}u = g \qquad \text{in } \Omega \tag{2.1a}$$

$$u = U_b \qquad \text{on } \Gamma, \tag{2.1b}$$

where the domain of the PDE is Ω, with boundary $\Gamma = \partial\Omega$. We will let $\mathcal{C}$ be a larger computational domain containing Ω. The method presented is not specific to two dimensions, but for this dissertation, we will take $\mathcal{C}$ to be a two-dimensional torus or periodic box. We will also assume Γ is a smooth one-dimensional curve that does not self-intersect.

The constraint method presented in this chapter has been developed in previous works [**35**, **47**, **59**], and in this dissertation, we will refer to this method as the Immersed Boundary Single Layer (IBSL) method. The rationale for this name will become clear in Chapter 4. It is presented here in order to introduce the notation and concepts that will be used in the presentation of the Immersed Boundary Double Layer (IBDL) method in subsequent chapters. Additionally, since one aim of this dissertation is to improve the efficiency of the IBSL method while preserving its robustness, we will make frequent comparisons to this method.

The first two sections of this chapter will focus on the IBSL method applied to scalar elliptic PDEs and the numerical implementation of the method. In Section 2.3, we then discuss the application of the method to Stokes and Brinkman equations, and in Section 2.4, we give some

(a) Exterior domain

(b) Interior domain
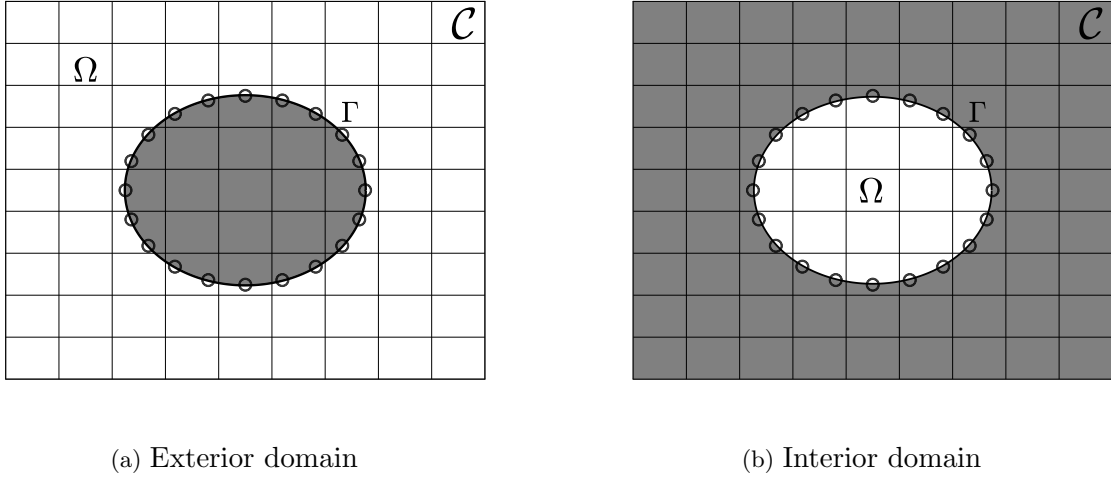
FIGURE 2.1. Diagram showing the two IB coordinate systems for an immersed structure Γ, embedded in a 2-D computational domain, $\mathcal{C}$. The PDE or physical domain, given by Ω, is shown in white, and the non-physical domain is shown in grey. Figure 2.1a shows an exterior PDE domain, and Figure 2.1b shows an interior PDE domain.

numerical results. Lastly, in Section 2.5, we discuss the benefits and limitations of the method in order to further motivate the IBDL method.

2.1 Mathematical description of IBSL method: scalar elliptic equations

In the Immersed Boundary method, one embeds the PDE domain Ω into a geometrically simple computational domain $\mathcal{C}$. The method then utilizes a Lagrangian coordinate system located on the immersed boundary, as well as an Eulerian coordinate system for the fluid. Figure 2.1 illustrates simple discretizations of these coordinate systems for two possible PDE domains.

In the original IB framework for interactions between fluid and elastic structures, the density of the force exerted by the structure on the fluid is represented by F, and it is supported on the boundary Γ. F is mapped to the Eulerian coordinate system through a convolution with a delta function. The resulting Eulerian force density is incorporated into the PDE, which is then solved in $\mathcal{C}$. With the elimination of the physical boundary, one can solve the discretized PDE on a regular grid using efficient solvers that are independent of the potentially complex geometry of Ω. After

7

the PDE is solved, the resulting solution u can then be mapped back to the boundary through another convolution with a delta function, and this gives the boundary values.

This dissertation will focus on the IB method for PDEs with *prescribed* boundary values, such as the PDE in Equation (2.1). In this so-called constraint formulation of the Immersed Boundary method, the force density F is an unknown Lagrange multiplier, used to enforce the boundary condition. For scalar elliptic problems, we will focus on the Helmholtz operator, $\mathcal{L} \equiv \Delta - k^2$, where $k = 0$ gives the Laplacian operator. We will use s and $\boldsymbol{x}$ as our Lagrangian and Eulerian coordinates, respectively, and $\boldsymbol{X}(s,t)$ gives the location of the boundary Γ. The inclusion of time, t, allows for a moving boundary, but for brevity, we will generally assume a fixed boundary and omit t from our equations. The equations for the continuous IB formulation of Equation (2.1) are then given by

$$\mathcal{L}\tilde{u}(\boldsymbol{x}) + \int_\Gamma F(s)\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))ds = \tilde{g}(\boldsymbol{x}) \qquad \text{in } \mathcal{C} \tag{2.2a}$$

$$\int_\mathcal{C} \tilde{u}(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))d\boldsymbol{x} = U_b(s) \qquad \text{on } \Gamma. \tag{2.2b}$$

In the above formulation, our unknowns are F and $\tilde{u}$. Note that $\tilde{u}(\boldsymbol{x})$ and $\tilde{g}(\boldsymbol{x})$ represent extensions of the solution u and the function g from Ω to the larger computational domain $\mathcal{C}$. Once the solution $\tilde{u}(x)$ is found in $\mathcal{C}$, the solution to the original PDE is then given by $\tilde{u}|_\Omega$. As such, we will drop the tilde notation on $u(x)$. On the other hand, we can define the extension of g by $\tilde{g} = g\chi_\Omega + g_e\chi_{\mathcal{C}\backslash\Omega}$, where χ is an indicator function. Since $\mathcal{C} \backslash \Omega$ is not in the PDE domain, there is flexibility in our choice for g_e. For example, one can simply use a smooth extension of g or use $g_e = 0$.

In the IB method, mappings between the coordinate systems are carried out with a regularized delta function δ_h, where the regularization lengthscale, h, is generally chosen to be on the order of the discretized grid point spacing. We can then view the first convolution as *spreading* the force density to the nearby grid points, and the second convolution as *interpolating* the solution values onto the boundary points. Therefore, let us define the spread and interpolation operators,

respectively, as

$$(SF)(\boldsymbol{x}) = \int_{\Gamma} F(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s,t))ds \tag{2.3a}$$

$$(S^*u)(s) = \int_{\mathcal{C}} u(\boldsymbol{x})\delta_h(\boldsymbol{x} - \boldsymbol{X}(s,t))d\boldsymbol{x}. \tag{2.3b}$$

These operators are adjoint in the sense that

$$\langle SF, u \rangle_{\mathcal{C}} = \langle F, S^*u \rangle_{\Gamma}, \tag{2.4}$$

where the inner products are the usual L^2 inner products on Ω and Γ, respectively. Discretization and numerical implementation will be discussed more in Section 2.2, but note that we will also use S and S^* to refer to the discretized versions of these operators. Similarly, let $\mathcal{L}$ represent our continuous or discretized elliptic differential operator. We then get the following system of equations.

$$\mathcal{L}u + SF = \tilde{g} \qquad \text{in } \mathcal{C} \tag{2.5a}$$

$$S^*u = U_b \qquad \text{on } \Gamma, \tag{2.5b}$$

or

$$\begin{pmatrix} \mathcal{L} & S \\ S^* & 0 \end{pmatrix} \begin{pmatrix} u \\ F \end{pmatrix} = \begin{pmatrix} \tilde{g} \\ U_b \end{pmatrix}. \tag{2.6}$$

As indicated in the beginning of the chapter, we will refer to the method described by Equation (2.5) as the Immersed Boundary Single Layer (IBSL) method.

2.2 Numerical implementation

2.2.1 Discretization of space and differential operators

Unless otherwise specified, we take the computational domain to be the two-dimensional periodic box, $\mathcal{C} = [-L/2, L/2]^2$, in order to make use of efficient finite difference and Fourier spectral methods

to solve Equation (2.5a). The computational domain is discretized with a regular Cartesian mesh with N points in each direction, giving us $\Delta x = \Delta y = L/N$. We then have N^2 nodes located at $\boldsymbol{x}_{i,j} = (-L/2 + i\Delta x, -L/2 + j\Delta y)$ for integers $0 \leq i, j \leq N - 1$. We can easily generalize this to a periodic box with different horizontal and vertical lengths and node spacings, but we make these choices for simplicity.

Since we will be looking at domains of various sizes, we rescale the discrete $L^1(\Omega)$ and $L^2(\Omega)$ norms by the area of Ω, denoted $|\Omega|$. Our refinement studies therefore use the discrete function norms given by

$$||u||_1 = \frac{\Delta x \Delta y}{|\Omega|} \sum_{i,j} u(\boldsymbol{x}_{i,j})\chi_\Omega \tag{2.7a}$$

$$||u||_2 = \left(\frac{\Delta x \Delta y}{|\Omega|} \sum_{i,j} (u(\boldsymbol{x}_{i,j}))^2 \chi_\Omega \right)^{1/2}. \tag{2.7b}$$

The IBSL method has flexibility in the choice for discretization of the differential operator. In this dissertation, we utilize both finite difference and Fourier spectral methods. In the case of finite differences, we will use the standard five-point, second-order accurate approximation for the Laplacian, given by

$$(\Delta u)_{i,j} \approx \frac{1}{\Delta x \Delta y}\Big(u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1} - 4u_{i,j} \Big), \tag{2.8}$$

where $u_{i,j} = u(\boldsymbol{x}_{i,j})$. For Fourier spectral methods, the Laplacian is given by

$$\Delta u(x) = -\mathcal{F}^{-1}(\mathbf{k} \cdot \mathbf{k}\mathcal{F}u(x)), \tag{2.9}$$

where $\mathcal{F}$ is the Discrete Fast Fourier Transform, $\mathcal{F}^{-1}$ is its inverse, and $\mathbf{k} = (k_1, k_2)$ gives the vector of wave-numbers.

2.2.2 Discretization of spread and interpolation operators

To discretize the boundary Γ, we use a set of N_{IB} boundary points, given by $\{X(s_i)\}_{i=1}^{N_{IB}}$. Since the locations of $\boldsymbol{X}(s_i)$ do not coincide with the grid points, we use the spread and interpolation operators, S and S^*, to map between the Lagrangian and Eulerian grids.

The spread operator maps a singular force distribution supported on Γ to an appropriate density on the Eulerian grid. It is formed as an approximate convolution with a regularized two-dimensional delta function. There is flexibility in the choice of the regularized delta function, but a suitable choice will satisfy certain properties [29,40,47]. In this dissertation, unless otherwise specified, we use the traditional Peskin four-point delta function [47], which in two dimensions is given by

$$\delta_h = \frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right), \tag{2.10}$$

for $h = \Delta x = \Delta y$ and

$$\phi(r) = \begin{cases} \frac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4|r|^2}\right) & 0 \le |r| \le 1 \\ \frac{1}{8}\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4|r|^2}\right) & 1 \le |r| \le 2 \\ 0 & |r| \ge 2. \end{cases} \tag{2.11}$$

Using this delta function, we define the continuous spread operator as

$$(SF)(\boldsymbol{x}) = \int_\Gamma F(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{2.12}$$

Then, discretizing the integral gives us the discrete operator,

$$(SF)(\boldsymbol{x}) = \sum_{i=1}^{N_{IB}} F(s_i)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s_i))\Delta s_i. \tag{2.13}$$

A graphical illustration of the spread operator is given in Figure 2.2. Note that this integral is respect to the parameter s. It is not necessary that s be an arclength parameter, nor is it necessary that the points be equally spaced, either physically or in terms of the parameter s. However, if we do space the points equally in one of these ways, we will get a spectrally accurate quadrature rule for a smooth periodic function F, defined on Γ. Unless otherwise specified, we will use arclength
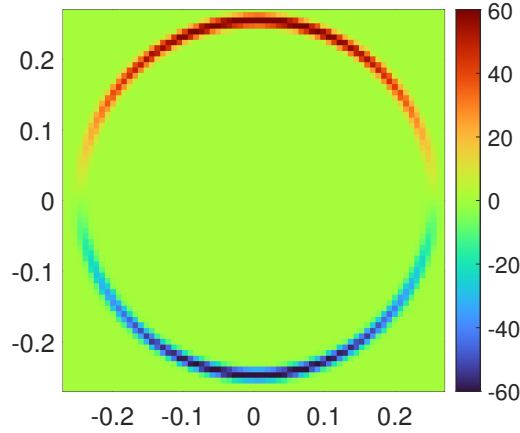
FIGURE 2.2. Illustration of the spread operator defined in Equation (2.13). The boundary Γ is a circle of radius 0.25 centered at the origin, and the spread operator is applied to the distribution $F = \sin\theta$. The boundary point spacing is given by $\Delta s \approx 2\Delta x$.

parametrization and equally spaced points with $\Delta s_i = L_{IB}/N_{IB}$, where L_{IB} is the length of the immersed boundary. Additionally, we will select N_{IB} such that $\Delta s \approx \alpha \Delta x$ for various values of α. As will be discussed further in Section 2.4, a commonly used point spacing for the IBSL method is $\Delta s \approx 2\Delta x$ [35]. If the exact values of Δs_i are unavailable, approximations can be made. In the case of an arclength parametrization, using $\Delta s_i \approx ||\boldsymbol{X}(s_{i+1}) - \boldsymbol{X}(s_i)||_2$ is sufficient.

We lastly look at the the interpolation operator, which maps a discrete function on the Eulerian grid to a discrete function supported on the Lagrangian grid. It is discretized as follows.

$$(S^*u)(s) = \int_{\mathcal{C}} u(\boldsymbol{x})\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))d\boldsymbol{x} \approx \sum_{x_{i,j}} u(\boldsymbol{x}_i)\delta_h(\boldsymbol{x}_i - \boldsymbol{X}(s))(\Delta x)(\Delta y), \tag{2.14}$$

where the sum is over the Cartesian gridpoints, and for a fixed s and a four-point delta function, the sum has at most 16 non-zero terms.

2.2.3   Solution to discrete system for invertible $\mathcal{L}$

We now look at the discretized system

$$\mathcal{L}u + SF = \tilde{g} \qquad \text{in } \mathcal{C} \tag{2.15a}$$

$$S^*u = U_b \qquad \text{on } \Gamma. \tag{2.15b}$$

12

In the case that the differential operator is invertible, such as for $\mathcal{L} = \Delta - k^2$, for $k \neq 0$, we can invert the operator to obtain

$$u = -\mathcal{L}^{-1}SF + \mathcal{L}^{-1}\tilde{g}. \tag{2.16}$$

Then, by applying the interpolation operator and using Equation (2.15b), we obtain

$$-(S^*\mathcal{L}^{-1}S)F = U_b - S^*\mathcal{L}^{-1}\tilde{g}. \tag{2.17}$$

We can therefore solve the saddle point problem in Equation (2.15) by first solving Equation (2.17) for F and then obtaining u from Equation (2.15a). Here, the operator that must be inverted, $-(S^*\mathcal{L}^{-1}S)$, is called the Schur complement of the system. We do not form the Schur complement matrix explicitly but instead create a routine that performs the action of the operator, and we solve for F with a Krylov method. Since the Schur complement is a symmetric operator, we use `minres`, to a tolerance of 10^{-8}, unless otherwise specified.

2.2.4 Solution to discrete system for Poisson equation

In the case that the differential operator is the periodic Laplacian, $\mathcal{L} = \Delta$, the solution method outlined in the previous section must be adjusted to account for the nullspace of $\mathcal{L}$. The boundary value problem given by

$$\Delta u = g \qquad \text{in } \Omega \tag{2.18a}$$

$$u = U_b \qquad \text{on } \Gamma \tag{2.18b}$$

has a unique solution in Ω, but, by using the IB framework, we embed the PDE into a periodic computational domain on which Δ is not invertible. Stein et al. [54] present a process for solving Equation (2.15) in this case. Since we will require a similar method when applying the IBDL method to Stokes equations, we present a version of this method here.

We first decompose the solution u as

$$u = u_0 + \bar{u}, \tag{2.19}$$

13

where u_0 has mean 0 on $\mathcal{C}$ and $\bar{u} = \frac{1}{|\mathcal{C}|} \int_{\mathcal{C}} u d\boldsymbol{x}$ is a constant giving the mean value of u on the computational domain. Note that a constant function spans the nullspace of the periodic Laplacian.

The PDE on the computational domain $\mathcal{C}$ that results from the IBSL formulation is then given by

$$\Delta u_0 + SF = \tilde{g}, \tag{2.20}$$

where we have used that $\Delta \bar{u} = 0$. To derive the solvability condition, we integrate Equation (2.20) over a general computational domain $\mathcal{C}$. Let us use $\partial \mathcal{C}$ as the boundary of $\mathcal{C}$ and $\boldsymbol{n}_{\mathcal{C}}$ as the unit normal on $\partial \mathcal{C}$, to distinguish it from $\boldsymbol{n}$, which we will later use as the unit normal on the immersed boundary Γ. We note that since S is defined using a regularized delta function, our functions are smooth, and we can use the divergence theorem. This gives us

$$\int_{\partial \mathcal{C}} \nabla u_0 \cdot \boldsymbol{n}_{\mathcal{C}} dl(\boldsymbol{x}) + \int_{\mathcal{C}} SF d\boldsymbol{x} = \int_{\mathcal{C}} \tilde{g} d\boldsymbol{x}, \tag{2.21}$$

where we use $dl(\boldsymbol{x})$ to denote an integral with respect to arclength. If we take $\mathcal{C}$ to be the periodic box used in this work, the first term disappears, and we get the following solvability condition for the periodic computational domain.

$$\int_{\mathcal{C}} SF d\boldsymbol{x} = \int_{\mathcal{C}} \tilde{g} d\boldsymbol{x} \tag{2.22}$$

Equation (2.22) then provides the constraint needed to solve for the additional unknown, $\bar{u}$. We can then discretize the previous equations and summarize the discrete IBSL system as

$$\Delta u_0 + SF = \tilde{g} \tag{2.23a}$$

$$S^* u_0 + S^* \bar{u} \mathbb{1}_{N^2} = U_b \tag{2.23b}$$

$$(\Delta x \Delta y) \mathbb{1}_{N^2}^{\mathsf{T}} SF = (\Delta x \Delta y) \mathbb{1}_{N^2}^{\mathsf{T}} \tilde{g}, \tag{2.23c}$$

where $\mathbb{1}_{N^2}$ denotes a vector of length N^2 consisting of all ones. Since we have ensured that the solvability constraint is satisfied, the solution to Equation (2.23a) can now be found, and the solution is unique up to an additive constant. We then let Δ_0^{-1} denote the operation that inverts the Laplacian by returning a solution with mean 0 on $\mathcal{C}$. Then we have $\Delta_0^{-1} \Delta u_0 = u_0$. Using this

to invert the Laplacian in Equation (2.23a) and then applying S^*, we get

$$S^* u_0 + S^* \Delta_0^{-1} SF = S^* \Delta_0^{-1} \tilde{g}. \tag{2.24}$$

Then, we can replace $S^* u_0$ using Equation (2.23b), and we get

$$U_b - S^* \bar{u} \mathbb{1}_{N^2} + S^* \Delta_0^{-1} SF = S^* \Delta_0^{-1} \tilde{g}. \tag{2.25}$$

We then have the following system with unknowns, F and $\bar{u}$:

$$\begin{pmatrix} -S^* \Delta_0^{-1} S & S^* \mathbb{1}_{N^2} \\ (\Delta x \Delta y) \mathbb{1}_{N^2}^\mathsf{T} S & 0 \end{pmatrix} \begin{pmatrix} F \\ \bar{u} \end{pmatrix} = \begin{pmatrix} U_b - S^* \Delta_0^{-1} \tilde{g} \\ (\Delta x \Delta y) \mathbb{1}_{N^2}^\mathsf{T} \tilde{g} \end{pmatrix}. \tag{2.26}$$

We can then use properties of the spread and interpolation operators to simplify this system. First, we note that the interpolation of a constant function will return a constant function on Γ, giving us

$$S^* \bar{u} \mathbb{1}_{N^2} = \bar{u} \mathbb{1}_{N_{IB}}, \tag{2.27}$$

where $\mathbb{1}_{N_{IB}}$ is a vector of length N_{IB} consisting of all ones. We can use this to change the operator acting on $\bar{u}$ in Equation (2.26). Next, letting $u = 1$ in the adjoint property given in Equation (2.4), we see that

$$\int_{\mathcal{C}} SF d\boldsymbol{x} = \int_{\Gamma} F ds, \tag{2.28}$$

or, in discretized terms,

$$(\Delta x \Delta y) \mathbb{1}_{N^2}^\mathsf{T} SF = \Delta s \mathbb{1}_{N_{IB}}^\mathsf{T} F. \tag{2.29}$$

The system can now be written as

$$\begin{pmatrix} -S^* \Delta_0^{-1} S & \mathbb{1}_{N_{IB}} \\ \Delta s \mathbb{1}_{N_{IB}}^\mathsf{T} & 0 \end{pmatrix} \begin{pmatrix} F \\ \bar{u} \end{pmatrix} = \begin{pmatrix} U_b - S^* \Delta_0^{-1} \tilde{g} \\ (\Delta x \Delta y) \mathbb{1}_{N^2}^\mathsf{T} \tilde{g} \end{pmatrix}. \tag{2.30}$$

Therefore, we can make this system symmetric by dividing the second equation by Δs, and our system becomes

$$
\begin{pmatrix} -S^*\Delta_0^{-1}S & \mathbb{1}_{N_{IB}} \\ \mathbb{1}_{N_{IB}}^\mathsf{T} & 0 \end{pmatrix} \begin{pmatrix} F \\ \bar{u} \end{pmatrix} = \begin{pmatrix} U_b - S^*\Delta_0^{-1}\tilde{g} \\ \frac{\Delta x \Delta y}{\Delta s}\mathbb{1}^\mathsf{T}\tilde{g} \end{pmatrix}.
\tag{2.31}
$$

Since the operator is symmetric, we again solve for F and $\bar{u}$ using `minres`. We can then obtain u_0 from Equation (2.23a) and add $\bar{u}$ to obtain our final solution,

$$
u = -\Delta_0^{-1}SF + \Delta_0^{-1}\tilde{g} + \bar{u}.
\tag{2.32}
$$

2.3 Mathematical description of IBSL method: Brinkman and Stokes equations

In this section, we apply the Immersed Boundary Single Layer method to boundary value problems of the form

$$
\mu\Delta\boldsymbol{u} - k^2\boldsymbol{u} - \nabla p = \boldsymbol{g} \qquad \text{in } \Omega
\tag{2.33a}
$$

$$
\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega
\tag{2.33b}
$$

$$
\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma,
\tag{2.33c}
$$

where in this dissertation, we will take $\boldsymbol{u}$ and $\boldsymbol{g}$ to be two-dimensional vector-valued functions. For $k = 0$, this is Stokes equation for viscous flow with zero Reynolds number, and $k \neq 0$ gives the Brinkman equation, which can be used for problems involving porous media [10]. The Brinkman equation will also arise in our implicit-explicit discretization of the Navier-Stokes equation, discussed in Chapter 6. The system of equations that results from applying the IBSL method to Equation (2.33) is

$$
\mu\Delta\boldsymbol{u} - k^2\boldsymbol{u} - \nabla p + S\boldsymbol{F} = \tilde{\boldsymbol{g}} \qquad \text{in } \mathcal{C}
\tag{2.34a}
$$

$$
\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \mathcal{C}
\tag{2.34b}
$$

$$
S^*\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma.
\tag{2.34c}
$$

16

We can see that the IBSL method generalizes in a straightforward way to the Stokes equation. Note that $\boldsymbol{F}$ is now a two-dimensional force density on the boundary Γ. Additionally, the spread and interpolation operators act element-wise on the discrete vector-valued functions. For example, $S\boldsymbol{F} = (S\boldsymbol{F}_1, S\boldsymbol{F}_2)$. In the case that $k = 0$, the nullspace of the periodic Laplacian will again require a more careful consideration similar to that seen in Section 2.2.4. We will therefore handle the two cases separately following a brief description of the discretization of the differential operators.

2.3.1 Discretization of differential operators

The discretization of space and the spread and interpolation operators is identical to that described in Section 2.2. We again use both finite difference and Fourier spectral methods for discretizing the differential operators. For Fourier spectral methods and two dimensions, we compute the derivative with multi-index $\alpha = (\alpha_1, \alpha_2)$ as

$$u^\alpha(\boldsymbol{x}) = \mathcal{F}^{-1}\left(i^{|\alpha|} k_1^{\alpha_1} k_2^{\alpha_2} \mathcal{F}u(x)\right), \tag{2.35}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ give the Discrete Fast Fourier Transform and its inverse, and $\mathbf{k} = (k_1, k_2)$ gives the vector of wave-numbers.

For finite difference methods, we use centered, second-order accurate approximations for first derivatives, and we again use the standard five-point, second-order accurate approximation for the Laplacian. However, as will be described in the following section, in order to solve for pressure, we will need to invert the operator given by $\nabla \cdot \nabla$, which will not correspond to the standard five-point Laplacian given in Equation (2.8). It instead corresponds to a Laplacian with a wider stencil, given by

$$(\nabla \cdot \nabla u)_{i,j} \approx \frac{1}{4\Delta x \Delta y}\left(u_{i-2,j} + u_{i,j-2} + u_{i+2,j} + u_{i,j+2} - 4u_{i,j}\right), \tag{2.36}$$

This will result in an operator with a four-dimensional nullspace of constants on each of 4 subgrids. Therefore, in the following section, when we say that the mean of the pressure will be fixed at 0, in the case of this finite difference method, the mean will be fixed at 0 on each of the 4 subgrids.

A staggered-grid, or MAC, discretization is a common way to avoid the de-coupling of these 4 subgrids, as well as to improve volume conservation in Immersed Boundary fluid applications [8, 24, 25, 26, 31]. In this dissertation, we do not make use of a staggered grid in order to maintain consistency between our implementation of the IBSL and IBDL methods, and as will be discussed further in Section 6.3.1, generalizing this discretization to the IBDL method is non-trivial and merits future exploration.

2.3.2 Solution to discrete system for Brinkman equation

The IBSL formulation of the Brinkman equation with Dirichlet boundary conditions is again given by

$$\mu\Delta\boldsymbol{u} - k^2\boldsymbol{u} - \nabla p + S\boldsymbol{F} = \tilde{\boldsymbol{g}} \qquad \text{in } \mathcal{C} \tag{2.37a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \mathcal{C} \tag{2.37b}$$

$$S^*\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma, \tag{2.37c}$$

or

$$\begin{pmatrix} \mathcal{L} & -\nabla & S \\ \nabla\cdot & 0 & 0 \\ S^* & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \\ \boldsymbol{F} \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{g}} \\ 0 \\ \boldsymbol{U}_b \end{pmatrix}, \tag{2.38}$$

where $\mathcal{L} = \mu\Delta - k^2\mathbb{I}$, and $\mathbb{I}$ is the identity operator.

Taking the divergence of Equation (2.37a) and using the incompressibility of $\boldsymbol{u}$, we get

$$-\Delta p + \nabla \cdot S\boldsymbol{F} = \nabla \cdot \tilde{\boldsymbol{g}}. \tag{2.39}$$

Equation (2.39) is solvable on the periodic domain, and we invert the periodic Laplacian with the operator Δ_0^{-1}, which was described in Section 2.2.4. We therefore fix its mean value on $\mathcal{C}$ to be 0. Completing this inversion, we get

$$p = -\Delta_0^{-1}\nabla \cdot (\tilde{\boldsymbol{g}} - S\boldsymbol{F}). \tag{2.40}$$

18

Using this expression for p in Equation (2.37a), we get

$$\mathcal{L}\boldsymbol{u} = (\tilde{\boldsymbol{g}} - S\boldsymbol{F}) - \nabla \Delta_0^{-1} \nabla \cdot (\tilde{\boldsymbol{g}} - S\boldsymbol{F}). \tag{2.41}$$

Let us denote the operator that projects onto divergence-free fields as $\mathbb{P} = \mathbb{I} - \nabla \Delta_0^{-1} \nabla \cdot$. Equation (2.41) then becomes

$$\mathcal{L}\boldsymbol{u} = \mathbb{P}(\tilde{\boldsymbol{g}} - S\boldsymbol{F}). \tag{2.42}$$

Inverting $\mathcal{L}$ and applying the interpolation operator S^*, we get the following equation for $\boldsymbol{F}$:

$$-\left(S^* \mathcal{L}^{-1} \mathbb{P} S \right) \boldsymbol{F} = \boldsymbol{U}_b - S^* \mathcal{L}^{-1} \mathbb{P} \tilde{\boldsymbol{g}}. \tag{2.43}$$

As we did in Section 2.2.3, we solve this equation for $\boldsymbol{F}$ using `minres`, and then we solve Equations (2.37a) and (2.37b) for the velocity $\boldsymbol{u}$ and pressure p.

2.3.3 Solution to discrete system for Stokes equation

In the case of Stokes equation, for which $k = 0$, we must again adjust the method of solution since the periodic Laplacian is not invertible. We can follow a process similar to that in Section 2.2.4 to find the solvability constraint for Equation (2.37a). We again begin by decomposing the solution $\boldsymbol{u}$ as

$$\boldsymbol{u} = \boldsymbol{u}_0 + \bar{\boldsymbol{u}}, \tag{2.44}$$

where $\boldsymbol{u}_0$ has mean 0 on $\mathcal{C}$, and $\bar{\boldsymbol{u}} = (\bar{u}, \bar{v})$ gives the mean values of the horizontal and vertical components of velocity, u and v, respectively. By taking $\mathcal{L} = \mu \Delta$ and integrating Equation (2.41) over a general computational domain $\mathcal{C}$, we get

$$\mu \int_{\mathcal{C}} \Delta \boldsymbol{u}_0 d\boldsymbol{x} = \int_{\mathcal{C}} \left(\tilde{\boldsymbol{g}} - S\boldsymbol{F} - \nabla \Delta_0^{-1} \nabla \cdot (\tilde{\boldsymbol{g}} - S\boldsymbol{F}) \right) d\boldsymbol{x}. \tag{2.45}$$

Using the divergence theorem where applicable, we get

$$\mu \int_{\partial\mathcal{C}} \nabla \boldsymbol{u}_0 \cdot \boldsymbol{n}_\mathcal{C} dl(\boldsymbol{x}) = \int_\mathcal{C} \tilde{\boldsymbol{g}} d\boldsymbol{x} - \int_\mathcal{C} S\boldsymbol{F} d\boldsymbol{x} - \nabla\Delta_0^{-1} \int_{\partial\mathcal{C}} \tilde{\boldsymbol{g}} \cdot \boldsymbol{n}_\mathcal{C} dl(\boldsymbol{x}) + \nabla\Delta_0^{-1} \int_{\partial\mathcal{C}} S\boldsymbol{F} \cdot \boldsymbol{n}_\mathcal{C} dl(\boldsymbol{x}).$$

(2.46)

If we assume Γ is away from the boundary of $\mathcal{C}$, the last integral on the right-hand-side vanishes due to the compact support of the integrand. If we take $\mathcal{C}$ to be a periodic box, the left-hand-side also vanishes. If the PDE domain Ω is an exterior domain, then $\tilde{\boldsymbol{g}}$ on $\partial\mathcal{C}$ is $\boldsymbol{g}$, the periodic forcing function in Equation (2.33a). If, on the other hand, Ω is an interior domain, since we have flexibility in choosing our extended function $\tilde{\boldsymbol{g}}$, we can let $\tilde{\boldsymbol{g}}\chi_{\mathcal{C}\setminus\Omega} = \boldsymbol{g}_e = 0$. In either of these cases, the third integral on the right-hand-side vanishes. Using that $\int_\mathcal{C} S\boldsymbol{F} d\boldsymbol{x} = \int_\Gamma \boldsymbol{F} ds$, we again get the solvability constraint

$$\int_\Gamma \boldsymbol{F} ds = \int_\mathcal{C} \tilde{\boldsymbol{g}} d\boldsymbol{x}.$$

(2.47)

Therefore, using a process similar to that of Section 2.2.4, we obtain the following system for unknowns $\boldsymbol{F}$ and $\bar{\boldsymbol{u}}$.

$$\begin{pmatrix} -\frac{1}{\mu} S^* \Delta_0^{-1} \mathbb{P} S & \mathbb{1}_{N_{IB}} \\ \mathbb{1}_{N_{IB}}^\mathsf{T} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{F} \\ \bar{\boldsymbol{u}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{U}_b - \frac{1}{\mu} S^* \Delta_0^{-1} \mathbb{P}\tilde{\boldsymbol{g}} \\ \frac{(\Delta x)^2}{\Delta s} \mathbb{1}^\mathsf{T}\tilde{\boldsymbol{g}} \end{pmatrix}.$$

(2.48)

We again solve this system using `minres`, and after obtaining $\boldsymbol{F}$ and $\bar{\boldsymbol{u}}$, we get $\boldsymbol{u}$ by solving Equation (2.37a) for $\boldsymbol{u}_0$ and then adding $\bar{\boldsymbol{u}}$ to obtain our final solution,

$$\boldsymbol{u} = \frac{1}{\mu}\Delta_0^{-1}\mathbb{P}\Big(\tilde{\boldsymbol{g}} - S\boldsymbol{F}\Big) + \bar{\boldsymbol{u}}.$$

(2.49)

2.4 IBSL method: results

In this section, we provide some numerical results for the Immersed Boundary Single Layer method presented in this chapter. We draw particular attention to the high number of iterations of `minres` required to solve the systems found in Equations (2.17), (2.26), (2.43), and (2.48). These saddle

point problems resulting from Dirichlet boundary conditions are poorly conditioned [6]. There-fore, as mentioned in the introduction, much work has been devoted to forming precondition-ers [12, 30, 35, 54]. In this section, we will see that without proper preconditioning, the iteration counts can become extremely large. With an eye toward time-dependent PDE, such as the Navier-Stokes equation, in which the saddle point system would be solved in each time step, this can be prohibitively time consuming. Additionally, preconditioning itself can be computationally ex-pensive, as it often involves inverting a dense matrix. This poor conditioning is one of the main motivations for the development of the Immersed Boundary Double Layer (IBDL) method, which is able to achieve the same order of accuracy as the IBSL method while avoiding these high iteration counts and therefore eliminating the need for preconditioning. This method will be introduced for scalar elliptic PDEs in Chapter 5 and for Stokes equation in Chapter 6. Comparable tests will be performed with the IBDL method to compare to the results presented here.

2.4.1 Helmholtz equation

Our first test problem is the PDE

$$\Delta u - u = 0 \qquad \text{in } \Omega \tag{2.50a}$$

$$u = \sin 2\theta \qquad \text{on } \Gamma, \tag{2.50b}$$

where Ω is the interior of a circle of radius 0.25, centered at the origin. The analytical solution is given by

$$u = \frac{I_2(r) \sin 2\theta}{I_2(0.25)}, \tag{2.51}$$

where I_2 is the first-kind modified Bessel function of order 2. Our computational domain here is the periodic box $[-0.5, 0.5]^2$, and we use equally spaced boundary points with $\Delta s \approx \alpha \Delta x$ for various values of α. We use a finite difference method, and the solutions are computed for grid sizes ranging from $N = 2^6$ to 2^{12}.

Iteration Counts - Circular Boundary				
Δx	$\Delta s \approx 2\Delta x$	$\Delta s \approx 1.5\Delta x$	$\Delta s \approx 1\Delta x$	$\Delta s \approx 0.75\Delta x$
2^{-6}	17	45	249	491
2^{-7}	36	57	691	752
2^{-8}	49	71	1233	3057
2^{-9}	61	110	1922	5829
2^{-10}	68	144	1936	8084
2^{-11}	95	234	4364	9335
2^{-12}	142	303	4535	10589

TABLE 2.1. Number of iterations of `minres`, with tolerance 10^{-8}, needed to solve the Helmholtz equation (2.50) with the IBSL method, using finite differences, on the periodic computational domain $[-0.5, 0.5]^2$, where the circular boundary of radius 0.25 has prescribed boundary values given by $U_b = \sin 2\theta$.

Table 2.1 gives the iteration counts for using `minres`, to a tolerance of 10^{-8}, without preconditioning, to solve Equation (2.50) with the IBSL method. We can see that the the iteration counts increase as we refine our grid spacing. These counts would be prohibitively large in the case of a time-dependent problem, in which this solve would be require at each time step. The source of these large iteration counts will be discussed in Chapters 3 and 4. We also see the iteration counts increase as we refine the spacing of our immersed boundary points relative to the grid. For this reason, in order to make the IBSL method practical, it is common to use $\Delta s \approx 2\Delta x$ and to make use of preconditioning [**12**, **30**, **35**, **54**]. Figure 2.3 demonstrates the first-order convergence of the solution, which is characteristic of the IBSL method. In Chapter 5, we present a method that maintains this order of convergence while avoiding these high iteration counts.

2.4.2 Brinkman equation

Our second test problem is the Brinkman PDE

$$\Delta \boldsymbol{u} - \boldsymbol{u} - \nabla p = \boldsymbol{g} \qquad \text{in } \Omega \tag{2.52a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{2.52b}$$

$$u = \boldsymbol{U}_b \qquad \text{on } \Gamma. \tag{2.52c}$$
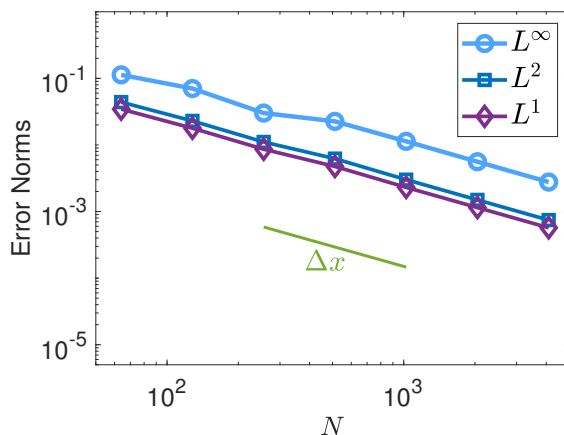
FIGURE 2.3. Refinement studies for solutions to Equation (2.50) found using the IBSL method with finite differences. The computational domain is the periodic box $[-0.5, 0.5]^2$, Ω is the interior of a circle of radius 0.25, and the prescribed boundary values are given by $U_b = \sin 2\theta$. The boundary point spacing is $\Delta s \approx 0.75\Delta x$, and the plot shows the L^1, L^2, and L^∞ norms of the absolute errors. In green, we denote the slope corresponding to first-order convergence.

We use an analytical solution given by

$$u = e^{\sin x} \cos y \tag{2.53a}$$

$$v = -\cos x \, e^{\sin x} \sin y \tag{2.53b}$$

$$p = e^{\cos y} \tag{2.53c}$$

to determine the boundary values, $\boldsymbol{U}_b = \boldsymbol{u}|_\Gamma$, and obtain the forcing function given by

$$\boldsymbol{g} = \begin{pmatrix} e^{\sin x} \cos y \, (\cos^2 x - \sin x - 2) \\ -\cos x \, e^{\sin x} \sin y \, (\cos^2 x - 3\sin x - 3) + \sin y \, e^{\cos y} \end{pmatrix}. \tag{2.54}$$

Here Ω is the interior of a circle of radius 0.75, centered at the origin, and the computational domain is the periodic box, $\mathcal{C} = [-1, 1]^2$. We use equally spaced boundary points with $\Delta s \approx \alpha \Delta x$ for various values of α. We use a Fourier spectral method, and the solutions are computed for grid sizes ranging from $N = 2^6$ to 2^{12}.

Table 5.1 gives the iteration counts for using `minres`, without preconditioning, to solve Equation (2.52) with the IBSL method. Note that for $\Delta s \approx 0.75\Delta x$, `minres` stagnated without reaching the tolerance of 10^{-8}, and the iteration counts in the last column give the points at which the method

23

Iteration Counts - Circular Boundary				
Δx	$\Delta s \approx 2\Delta x$	$\Delta s \approx 1.5\Delta x$	$\Delta s \approx 1\Delta x$	$\Delta s \approx 0.75\Delta x$
2^{-5}	123	296	2558	16190 stag.
2^{-6}	185	472	5578	47863 stag.
2^{-7}	257	678	14438	95009 stag.
2^{-8}	351	825	17475	68861 stag.
2^{-9}	478	1137	17789	62697 stag.
2^{-10}	622	1620	20333	75910 stag.
2^{-11}	821	2180	27467	79699 stag.

TABLE 2.2. Number of iterations of `minres`, with tolerance 10^{-8}, needed to solve Equation (2.52) with the IBSL method, using a Fourier spectral method, on the periodic computational domain $[-1, 1]^2$. The word *stag.* is used to indicate that the method stagnated at this number of iterations without converging to the desired tolerance.

stagnated. We can again see that the iteration counts increase as we refine our grid spacing or the spacing of our immersed boundary points relative to the grid. Figure 2.4a provides a plot of the solution on Ω for $N = 2^{10}$ and $\Delta s = 1.5\Delta x$. The velocity vectors are plotted every 2^6 meshwidths, and the color shows the pressure. As discussed in Section 2.3.2, the pressure is unique only up to an additive constant. We therefore solve for the pressure that has mean 0 on $\mathcal{C}$. We then find the difference between the computed pressure and analytical solution in Equation (2.53c) at the origin, and we add this constant to the computed pressure for the plot. Figures 2.4b-2.4c demonstrate the first-order convergence of the velocity solutions. We return to this problem in Chapter 6 to compare these results to those of the IBDL method.

2.5   Discussion

The Immersed Boundary Single Layer method is extremely robust and flexible. One can use the discretization and PDE solver of their choice without needing it to conform to the geometry of the boundary. It also has some key advantages over the boundary integral methods that will be discussed in the next chapter. For example, it can be easily adapted to various exterior domains, such as the periodic domain we utilize in this dissertation. It can also be easily used with non-homogeneous PDE without greatly increasing the computing work required, and the solution can be calculated on an entire domain mesh very quickly. Additionally, this method can be used for Navier-Stokes and other nonlinear problems by treating the nonlinearity explicitly in time. For

(a) Solution
S

(b) u

(c) v
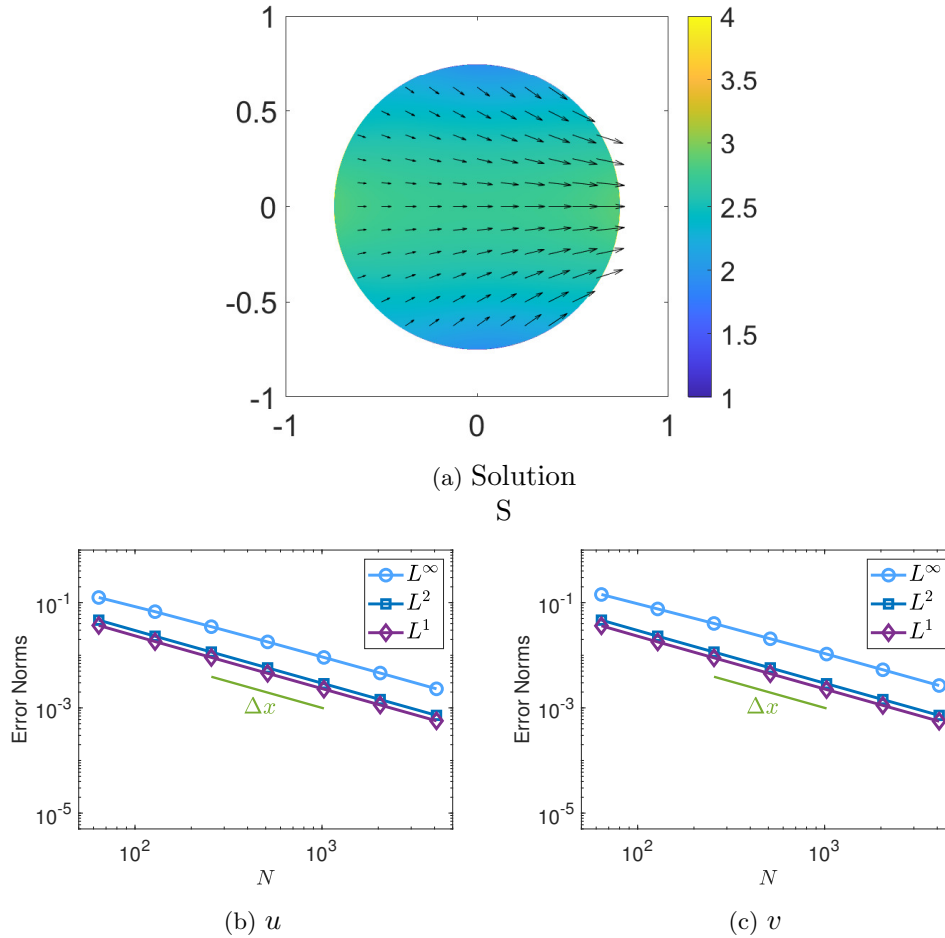
FIGURE 2.4. Solution plot and refinement studies, showing absolute errors, for solution to Equation (2.52) found using the IBSL method. The computational domain is the periodic box $[-1, 1]^2$, and the boundary point spacing is $\Delta s \approx 1.5\Delta x$. Figure 2.4a shows the velocity solution with vectors and pressure solution with color. It is found using the grid size $N = 2^{10}$.

these reasons, we strive to maintain the overall Immersed Boundary framework in our new method. However, we seek to improve on two areas. Firstly, the IBSL method cannot be used for PDEs with Neumann boundary conditions, and secondly, the conditioning of the Schur complement can make this method inefficient, especially in time-dependent problems. Therefore, we seek to remedy these two areas by reformulating the method using concepts from boundary integral equations.

Chapter 3

Boundary integral equations

Boundary integral methods can also be used to solve the PDEs in Chapter 2 by reformulating boundary value problems as integral equations using the appropriate Green's functions. In this chapter, we give an introduction to these boundary integral equations and the relevant properties that we will use when analyzing the IBSL method and developing the IBDL method. We begin by looking at boundary integral equations for Laplace's equation, and then we do the same for Stokes equation. Much of the material in this chapter can be found in boundary integral textbooks, such as [50] and [51], but we present it here in order to establish notation and motivation for the form of the IBDL method.

3.1   Green's functions

We begin by looking at Laplace's equation,

$$\Delta u = 0 \quad \text{in } \Omega. \tag{3.1}$$

A Green's function for this PDE is a function $G(\boldsymbol{x}, \boldsymbol{x}_0)$ that satisfies

$$\Delta G(\boldsymbol{x}, \boldsymbol{x}_0) = -\delta(\boldsymbol{x} - \boldsymbol{x}_0) \quad \text{in } \Omega, \tag{3.2}$$

where $\boldsymbol{x}_0$ is the location of a singular point force. One must also specify boundary conditions to determine a particular Green's function. For example, the free-space Green's function for an infinite solution domain in two-dimensions is

$$G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) = -\frac{1}{2\pi} \ln |\boldsymbol{x} - \boldsymbol{x}_0|. \tag{3.3}$$

This Green's function would be sufficient for solving boundary value problems for interior PDE domains. However, the exterior domains explored in this dissertation are not in free-space, but in a periodic box. However, we can use this function to analyze behavior near the singular point because all Green's functions for Laplace's equation exhibit the same singular behavior as $\boldsymbol{x} \longrightarrow \boldsymbol{x}_0$. In fact, any Green's function can be decomposed as

$$G(\boldsymbol{x}, \boldsymbol{x}_0) = G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) + H^C(\boldsymbol{x}, \boldsymbol{x}_0), \tag{3.4}$$

where $H^C(\boldsymbol{x}, \boldsymbol{x}_0)$ is a complementary harmonic function that is non-singular on Ω [51].

We next derive an integral property of the Green's function derivative that will be important for obtaining our integral equations. For convenience, let our problem domain Ω be an interior domain, unless otherwise specified, with boundary Γ. In this chapter, when we omit explicit discussion of an exterior domain, the steps and conclusions follow similarly as for the interior domain. Taking $\boldsymbol{x}_0 \notin \Omega$, we can integrate Equation (3.2) about Ω, and use the divergence theorem to get

$$0 = \int_{\Omega} \Delta G(\boldsymbol{x}, \boldsymbol{x}_0) d\boldsymbol{x} = \int_{\Gamma} \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}), \tag{3.5}$$

where $dl(\boldsymbol{x})$ denotes an integral with respect to arclength, and $\boldsymbol{n}$ is the unit normal pointing out of Ω, regardless of whether Ω is an exterior or interior domain.

If, on the other hand, $\boldsymbol{x}_0 \in \Omega$, since the Green's function is not continuously differentiable at $\boldsymbol{x} = \boldsymbol{x}_0$, in order to use the divergence theorem, we can remove a circle centered at $\boldsymbol{x}_0$ with sufficiently small radius ϵ and use the divergence theorem on the region $\Omega' = \Omega \setminus B_\epsilon(\boldsymbol{x}_0)$. Figure 3.1a gives an illustration. Using the decomposition of G in Equation (3.4), we then have

$$0 = \int_{\Omega'} \Delta G(\boldsymbol{x}, \boldsymbol{x}_0) d\boldsymbol{x} = \int_{\Omega'} \Delta G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) d\boldsymbol{x} + \int_{\Omega'} \Delta H^C(\boldsymbol{x}, \boldsymbol{x}_0) d\boldsymbol{x} = \int_{\Omega'} \Delta G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) d\boldsymbol{x},$$

$$\tag{3.6}$$

where we have used that H^C is a harmonic function. Using the divergence theorem then gives us

$$0 = \int_{\Gamma} \nabla G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) + \int_{\partial B_\epsilon} \nabla G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}), \tag{3.7}$$

(a) $\boldsymbol{x}_0 \in \Omega$ (b) $\boldsymbol{x}_0 \in \Gamma$
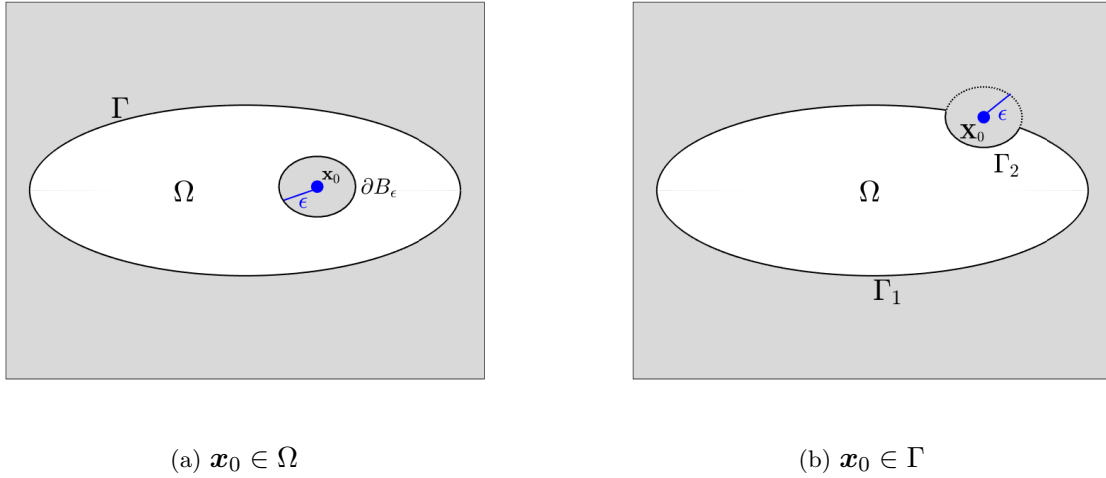
FIGURE 3.1. Figure 3.1a illustrates the removal of ball $B_\epsilon(\boldsymbol{x}_0)$ from the PDE domain Ω for $\boldsymbol{x}_0 \in \Omega$ in order to use the divergence theorem in Equation (3.7). The resulting domain, Ω', excludes $\boldsymbol{x}_0$ and has the boundary $\Gamma \cup \partial B_\epsilon$. Figure 3.1b illustrates the removal of $\Omega \cap B_\epsilon(\boldsymbol{X}_0)$ from Ω for $\boldsymbol{X}_0 \in \Gamma$ The resulting domain, Ω', excludes $\boldsymbol{X}_0$ and has the boundary $\Gamma_1 \cup \Gamma_2$.

where both unit normal vectors are those that point out of Ω'. The free-space Green's function derivative is

$$\nabla G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) = -\frac{1}{2\pi |\boldsymbol{x} - \boldsymbol{x}_0|^2} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}. \tag{3.8}$$

Additionally, the unit normal on ∂B_ϵ is

$$\boldsymbol{n} = -\frac{\boldsymbol{x} - \boldsymbol{x}_0}{|\boldsymbol{x} - \boldsymbol{x}_0|}. \tag{3.9}$$

Then the second integral is

$$\int_{\partial B_\epsilon} \nabla G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) = \frac{1}{2\pi} \int_{\partial B_\epsilon} \frac{|\boldsymbol{x} - \boldsymbol{x}_0|^2}{|\boldsymbol{x} - \boldsymbol{x}_0|^3} dl(\boldsymbol{x}) = \frac{1}{2\pi\epsilon} \int_{\partial B_\epsilon} dl(\boldsymbol{x}) = 1. \tag{3.10}$$

Then Equation (3.7) becomes

$$-1 = \int_\Gamma \nabla G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}). \tag{3.11}$$

28

In order to find this property for the general Green's function, note that, using that H^C is harmonic and non-singular on Ω, we have

$$0 = \int_\Omega \Delta H^C(\boldsymbol{x}, \boldsymbol{x}_0) d\boldsymbol{x} = \int_\Gamma \nabla H^C(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}). \tag{3.12}$$

Therefore, we have

$$\int_\Gamma \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) = \int_\Gamma \nabla G^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) + \int_\Gamma \nabla H^C(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) = -1,$$
$$\tag{3.13}$$

for $\boldsymbol{x}_0 \in \Omega$.

We next take the singularity to be on the boundary Γ. Since in the IBSL method, we denoted points on the boundary with a capital $\boldsymbol{X}(s)$, in order to maintain consistency and also distinguish from the case when $\boldsymbol{x}_0$ is not on the boundary, we will call this singularity $\boldsymbol{X}_0$. In order to use the divergence theorem, we then remove from Ω a circle centered at $\boldsymbol{X}_0$ with radius ϵ and again call this new region Ω'. An illustration of this is given in Figure 3.1b. Let Γ_1 denote the region of Γ that borders Ω' (i.e. the portion of Γ that excludes the region near $\boldsymbol{X}_0$), and let $\Gamma_2 = \partial\Omega' \setminus \Gamma_1$. Then, since $\boldsymbol{X}_0 \notin \Omega'$, we can use the divergence theorem to get

$$0 = \int_{\Gamma_1 \cup \Gamma_2} \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) = \int_{\Gamma_1} \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) + \int_{\Gamma_2} \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}).$$
$$\tag{3.14}$$

We can again use the decomposition of G into the free-space Green's function and the complementary harmonic function in order to use the free-space Green's function derivative. For brevity, we omit the details as they follow the same process as used for $\boldsymbol{x}_0 \in \Omega$. We can therefore use the form of the free-space Green's function and note that since Γ is smooth, as $\epsilon \longrightarrow 0$, the second integral approaches the integral over a semicircle of radius ϵ. Then, this integral simplifies to $1/2$, and we get

$$-\frac{1}{2} = \int_{\Gamma_1} \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}). \tag{3.15}$$

Let us define the principal-value integral, denoted PV, as the integral for which $\boldsymbol{X}_0$ is placed exactly on Γ. It is computed by integrating over $\Gamma \setminus B_\epsilon(\boldsymbol{X}_0)$, in which a disk around $\boldsymbol{X}_0$ has been removed from the boundary. The PV integral then comes from taking the limit as ϵ goes to 0. If we take $\epsilon \longrightarrow 0$ for the definition of Γ_1 in Equation (3.15), we get exactly this principal-value integral, so we have

$$-\frac{1}{2} = \int_\Gamma^{PV} \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}). \tag{3.16}$$

Therefore, to summarize Equations (3.5), (3.13), and (3.16), we can give the following integral identity for the Green's function derivative:

$$\int_\Gamma \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) = \begin{cases} -1, & \text{if } \boldsymbol{x}_0 \in \Omega \\ -\frac{1}{2}, & \text{if } \boldsymbol{x}_0 \in \Gamma \ , \\ 0, & \text{if } \boldsymbol{x}_0 \notin \bar{\Omega} \end{cases} \tag{3.17}$$

where $\bar{\Omega} = \Omega \cup \Gamma$. We can also rewrite Equation (3.17) as

$$\int_\Gamma \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) = \begin{cases} \int_\Gamma^{PV} \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) - \frac{1}{2} & \text{if } \boldsymbol{x}_0 \in \Omega \\ \int_\Gamma^{PV} \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) + \frac{1}{2} & \text{if } \boldsymbol{x}_0 \notin \bar{\Omega}, \end{cases} \tag{3.18}$$

and this version of the property will be used in Section 3.3.

3.2   Integral representations for solutions to Laplace's equation

In this section, we derive integral representations for solutions to Equation (3.1). We begin with Green's second identity. When applied to the Green's function and a solution to the PDE, u, the identity simplifies to

$$u(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{x}_0) = \nabla \cdot \Big(G(\boldsymbol{x}, \boldsymbol{x}_0)\nabla u(\boldsymbol{x}) - u(\boldsymbol{x})\nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \Big). \tag{3.19}$$

Let $\boldsymbol{x}_0 \in \Omega$. Then, in order to use the divergence theorem, we can again remove from Ω a ball of radius ϵ, centered at the singularity. By following similar steps to those in the previous section, we

can then get an integral representation of u given by

$$u(\boldsymbol{x}_0) = \int_\Gamma G(\boldsymbol{x}, \boldsymbol{x}_0) \nabla u(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) - \int_\Gamma u(\boldsymbol{x}) \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}), \qquad (3.20)$$

where $\boldsymbol{n}$ is the unit normal of Γ, pointing out of Ω.

The first term of Equation (3.20) is called a *single layer potential*, which can be viewed as a distribution of point forces on the boundary, and the second term is called a *double layer potential*, which can be viewed as a distribution of point force dipoles on the boundary. Notice that the strength of the single layer potential is given by the boundary distribution of the normal derivative of u, and the strength of the double layer potential is given by the boundary values of u.

There are, however, other integral representations of the solution u in Ω. We can, for instance, represent u solely with a single layer potential or a double layer potential. Let us consider two solutions: u^{Int} that satisfies the PDE on Ω and u^{Ext} that satisfies the PDE on Ω^{Ext}, the region exterior to Γ. Let these two solutions share the same values on Γ. In order to find a single layer representation for $u^{Int}(\boldsymbol{x}_0)$ for $\boldsymbol{x}_0 \in \Omega$ we need two equations. Firstly, we use Equation (3.20) for u^{Int}. Secondly, by the same process that gave us Equation (3.20), we can get a similar equation for u^{Ext}, where the corresponding normal vector points outward from Ω^{Ext}, and the left-hand side vanishes because $\boldsymbol{x}_0 \notin \Omega^{Ext}$. By negating the expression in order to use the same unit normal as that in Equation (3.20), we get the following equation for u^{Ext}:

$$0 = -\int_\Gamma G(\boldsymbol{x}, \boldsymbol{x}_0) \nabla u^{Ext}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) + \int_\Gamma u^{Ext}(\boldsymbol{x}) \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}). \qquad (3.21)$$

Then, by adding Equation (3.20), for u^{Int}, and Equation (3.21), we get the *generalized single layer integral representation* for u^{Int} given by

$$u^{Int}(\boldsymbol{x}_0) = \int_\Gamma \sigma(\boldsymbol{x}) G(\boldsymbol{x}, \boldsymbol{x}_0) dl(\boldsymbol{x}), \qquad (3.22)$$

where the strength of the single layer potential is given by the jump in normal derivatives across the boundary, or

$$\sigma = \nabla(u^{Int} - u^{Ext}) \cdot \boldsymbol{n}\Big|_\Gamma. \qquad (3.23)$$

If we consider two solutions, u^{Int} and u^{Ext} that, instead of matching values on the boundary, match normal derivatives on the boundary, we get the *generalized double layer integral representation* for u^{Int} given by

$$u^{Int}(\boldsymbol{x}_0) = \int_\Gamma \gamma(\boldsymbol{x})\nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x})dl(\boldsymbol{x}), \tag{3.24}$$

where the strength of the double layer potential is given by the jump in solution values across the boundary, or

$$\gamma = \left. (u^{Ext} - u^{Int}) \right|_\Gamma. \tag{3.25}$$

3.3 Integral equations for Laplace's equation with Dirichlet boundary conditions

In this section, we introduce the integral equations that arise from using the integral representations from the previous section to solve the Dirichlet Laplace's boundary value problem given by

$$\Delta u = 0 \qquad \text{in } \Omega \tag{3.26a}$$

$$u = U_b \qquad \text{on } \Gamma. \tag{3.26b}$$

In order to use our boundary data, we must take $\boldsymbol{x}_0 \longrightarrow \Gamma$, and we will again denote a point on the boundary as $\boldsymbol{X}_0$. The single layer potential seen in Equation (3.22) is continuous as $\boldsymbol{x}_0$ approaches and crosses the boundary [51], but the double layer potential is not. We therefore seek the limit of Equation (3.24) for $\boldsymbol{x}_0 \longrightarrow \Gamma$. We can first split up the integral using the potential strength at $\boldsymbol{x}_0$, or

$$\lim_{\boldsymbol{x}_0 \to \Gamma} \int_\Gamma \gamma(\boldsymbol{x})\nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x})dl(\boldsymbol{x})$$

$$= \lim_{\boldsymbol{x}_0 \to \Gamma} \left(\int_\Gamma \Big(\gamma(\boldsymbol{x}) - \gamma(\boldsymbol{x}_0)\Big)\nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x})dl(\boldsymbol{x}) + \gamma(\boldsymbol{x}_0)\int_\Gamma \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x})dl(\boldsymbol{x}) \right).$$
$$\tag{3.27}$$

Then we can use the identity given in Equation (3.18) to rewrite this as

$$\lim_{\boldsymbol{x}_0 \to \Gamma} \left(\int_\Gamma \Big(\gamma(\boldsymbol{x}) - \gamma(\boldsymbol{x}_0) \Big) \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) \right.$$
$$\left. + \gamma(\boldsymbol{x}_0) \left(\int_\Gamma^{PV} \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) - \frac{1}{2} \right) \right). \quad (3.28)$$

Finally, we distribute, take the limit, and recombine the integrals to obtain

$$\lim_{\boldsymbol{x}_0 \to \Gamma} \int_\Gamma \gamma(\boldsymbol{x}) \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) = \int_\Gamma^{PV} \gamma(\boldsymbol{x}) \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) - \frac{1}{2} \gamma(\boldsymbol{X}_0). \quad (3.29)$$

Using the limit given in Equation (3.29) for the double layer representation and the boundary condition in Equation (3.26b), the single and double layer integral equations are given by

$$U_b(\boldsymbol{X}_0) = \int_\Gamma \sigma(\boldsymbol{x}) G(\boldsymbol{x}, \boldsymbol{X}_0) dl(\boldsymbol{x}), \quad (3.30)$$

$$U_b(\boldsymbol{X}_0) = \int_\Gamma^{PV} \gamma(\boldsymbol{x}) \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) - \frac{1}{2} \gamma(\boldsymbol{X}_0), \quad (3.31)$$

where the unknown quantities in Equations (3.30) and (3.31) are the potential strengths, σ and γ, respectively. Note that the integral equations for a Dirichlet Helmholtz boundary value problem are the same as those for Laplace's equation, with the exception of the Green's function.

3.4 Green's functions for Stokes equation

In this section, we look at Stokes equation for incompressible viscous flow,

$$\mu \Delta \boldsymbol{u} - \nabla p = \nabla \cdot \boldsymbol{\sigma} = 0 \qquad \text{in } \Omega \quad (3.32a)$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \qquad \text{in } \Omega, \quad (3.32b)$$

where $\boldsymbol{u}$ is velocity, p is pressure, $\boldsymbol{\sigma}$ is the stress tensor, and μ is the viscosity of the fluid. We will again focus on the case where $\boldsymbol{u}$ is a two-dimensional vector-valued function. In this case, we define

the Green's function to be the solution to the singularly forced Stokes equation,

$$\mu \Delta \boldsymbol{u}^G - \nabla p^G = \nabla \cdot \boldsymbol{\sigma}^G = -\delta(\boldsymbol{x} - \boldsymbol{x}_0)\mathbf{c} \qquad \text{in } \Omega \tag{3.33a}$$

$$\nabla \cdot \boldsymbol{u}^G = 0 \qquad\qquad\qquad\qquad \text{in } \Omega, \tag{3.33b}$$

where $\mathbf{c}$ is a 2-D constant vector, and we use the superscript G to distinguish the singularly forced solution from a general solution to Stokes equation. Then, letting $\boldsymbol{G}$ be the Green's function and $\boldsymbol{p}$ and $\boldsymbol{T}$ be the pressure vector and the stress tensor corresponding to the Green's function, we can write the solution to Equation (3.33) as

$$u_i(\boldsymbol{x}) = \frac{1}{\mu} G_{ij}(\boldsymbol{x}, \boldsymbol{x}_0) c_j \tag{3.34a}$$

$$p(\boldsymbol{x}) = p_i(\boldsymbol{x}, \boldsymbol{x}_0) c_j \tag{3.34b}$$

$$\sigma_{ik}(\boldsymbol{x}) = T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0) c_j. \tag{3.34c}$$

Note that we use Einstein notation to indicate summation over repeated indices. We will use this notation often for sections on Stokes equation. As discussed in Section 3.1, the Green's function depends on the boundary conditions enforced on the solution domain, but the free-space Green's function provides the form of any such Green's function near the singular point, so we can use it to establish the properties of the Green's function. In free-space, the Green's function and corresponding stress tensor are called the Stokeslet and Stresslet, respectively. If we use $\hat{\boldsymbol{x}} = (\boldsymbol{x} - \boldsymbol{x}_0)$, then in two dimensions, the Stokeslet, Stresslet, and correponsing pressure vector are given by

$$G_{ij}^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) = -\frac{\delta_{ij} \ln |\hat{\boldsymbol{x}}|}{4\pi} + \frac{\hat{x}_i \hat{x}_j}{4\pi |\hat{\boldsymbol{x}}|^2} \tag{3.35a}$$

$$p_i^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) = \frac{\hat{x}_i}{2\pi |\hat{\boldsymbol{x}}|^2} \tag{3.35b}$$

$$T_{ijk}^{FS}(\boldsymbol{x}, \boldsymbol{x}_0) = -\frac{\hat{x}_i \hat{x}_j \hat{x}_k}{\pi |\hat{\boldsymbol{x}}|^4}, \tag{3.35c}$$

where δ_{ij} is the Kronecker delta, which is 1 when $i = j$ and 0 otherwise. A general Green's function, stress tensor, and pressure vector can be decomposed into the free-space functions and

34

complementary functions that are non-singular in Ω. Near the singularity, all Green's functions therefore exhibit the same leading order behavior as the free-space Green's function.

We will now derive an identity for the stress tensor $\boldsymbol{T}$ that is similar to the identity for the Green's function derivative given in Equation (3.18). We start with the singularly forced Stokes equation and use the expression for $\boldsymbol{\sigma}$ to get

$$\frac{\partial}{\partial x_k}\Big(T_{ijk}c_j\Big) = -\delta(\boldsymbol{x} - \boldsymbol{x}_0)c_i. \tag{3.36}$$

In two dimensions, this is two equations, but by reorganizing each one to eliminate the arbitrary constants, c_i, we get four equations summarized by

$$\frac{\partial T_{ijk}}{\partial x_k} = -\delta_{ij}\delta(\boldsymbol{x} - \boldsymbol{x}_0). \tag{3.37}$$

For $\boldsymbol{x}_0 \notin \Omega$, we can use the divergence theorem to get

$$\int_\Gamma T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) = 0. \tag{3.38}$$

Then for the case of $\boldsymbol{x}_0 \in \Omega$, as we did in Section 3.1, we can use the altered domain in Figure 3.1a and a decomposition of the stress tensor to find that

$$\int_\Gamma T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) = -\delta_{ij}. \tag{3.39}$$

Lastly, we look at the case in which the singularity, $\boldsymbol{X}_0$ is on the boundary Γ. We use the domain in Figure 3.1b, created by removing from Ω a circle centered at $\boldsymbol{X}_0$ with radius ϵ. Then, since $\boldsymbol{X}_0 \notin \Omega'$, we can use the divergence theorem to get

$$0 = \int_{\Gamma_1 \cup \Gamma_2} T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) = \int_{\Gamma_1} T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) + \int_{\Gamma_2} T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}). \tag{3.40}$$

Then, by using the decomposition of the stress tensor, we can again use the form of the Stresslet to calculate the integral. Additionally, as $\epsilon \longrightarrow 0$, the second integral approaches the integral over

a semicircle. Then this integral simplifies to

$$-\frac{1}{\pi\epsilon^3}\int_0^\pi x_i x_j x_k n_k d\theta = \frac{1}{\pi\epsilon^2}\int_0^\pi x_i x_j d\theta = \frac{1}{2}\delta_{ij}. \tag{3.41}$$

We therefore get the principal-value integral,

$$-\frac{1}{2}\delta_{ij} = \int_\Gamma^{PV} T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}), \tag{3.42}$$

and we have the following integral identity for the stress tensor:

$$\int_\Gamma T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) = \begin{cases} -\delta_{ij}, & \text{if } \boldsymbol{x}_0 \in \Omega \\ -\frac{1}{2}\delta_{ij}, & \text{if } \boldsymbol{x}_0 \in \Gamma \\ 0, & \text{if } \boldsymbol{x}_0 \notin \bar{\Omega}, \end{cases} \tag{3.43}$$

where $\bar{\Omega} = \Omega \cup \Gamma$. This property is again used to derive the integral equations presented in Section 3.6.

3.5 Integral representations for solutions to Stokes equation

3.5.1 Full integral representation

In this section, we derive the full integral representation for a solution to Equation (3.32). We begin by deriving the reciprocal relation for Stokes equation, which plays the same role that Green's second identity did in Section 3.2. Let $\boldsymbol{u}$, p, $\boldsymbol{\sigma}$ and $\boldsymbol{u}'$, p', $\boldsymbol{\sigma}'$ define two solutions to Equation (3.32). Then the product rule gives us

$$u_i'\frac{\partial\sigma_{ij}}{\partial x_j} = \frac{\partial(u_i'\sigma_{ij})}{\partial x_j} - \sigma_{ij}\frac{\partial u_i'}{\partial x_j} \tag{3.44}$$

Next, we use

$$\sigma_{ij} = -\delta_{ij}p + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{3.45}$$

to replace σ_{ij} in the second term of Equation (3.44). Then we simplify and use that $\nabla \cdot \boldsymbol{u}' = 0$ to get

$$u_i' \frac{\partial \sigma_{ij}}{\partial x_j} = \frac{\partial(u_i' \sigma_{ij})}{\partial x_j} - \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{\partial u_i'}{\partial x_j}. \tag{3.46}$$

Switching the flows to form a similar expression for $u_i(\partial \sigma_{ij}'/\partial x_j)$ and subtracting them, we get

$$u_i' \frac{\partial \sigma_{ij}}{\partial x_j} - u_i \frac{\partial \sigma_{ij}'}{\partial x_j} = \frac{\partial}{\partial x_j}\left(u_i' \sigma_{ij} - u_i \sigma_{ij}' \right). \tag{3.47}$$

Lastly, since they are both solutions to Stokes equations, $\sigma_{ij}/\partial x_j = \sigma_{ij}'/\partial x_j = 0$, and we get the reciprocal relation:

$$0 = \frac{\partial}{\partial x_j}\left(u_i' \sigma_{ij} - u_i \sigma_{ij}' \right). \tag{3.48}$$

Now, instead of using two solutions to Equation (3.32), we replace the second flow with the flow from the singularly forced Stokes equation. Then Equation (3.47) becomes

$$u_i^G \frac{\partial \sigma_{ij}}{\partial x_j} - u_i \frac{\partial \sigma_{ij}^G}{\partial x_j} = \frac{\partial}{\partial x_j}\left(u_i^G \sigma_{ij} - u_i \sigma_{ij}^G \right), \tag{3.49}$$

or

$$u_i \delta(\boldsymbol{x} - \boldsymbol{x}_0) c_i = \frac{\partial}{\partial x_j}\left(\frac{1}{\mu} G_{im} c_m \sigma_{ij} - u_i T_{imj} c_m \right). \tag{3.50}$$

This is a scalar equation, but by reorganizing to eliminate the arbitrary constants, c_i, we get two equations summarized by

$$u_i(\boldsymbol{x}) \delta(\boldsymbol{x} - \boldsymbol{x}_0) = \frac{\partial}{\partial x_j}\left(\frac{1}{\mu} G_{ki}(\boldsymbol{x}, \boldsymbol{x}_0) \sigma_{kj}(\boldsymbol{x}) - u_k(\boldsymbol{x}) T_{kij}(\boldsymbol{x}, \boldsymbol{x}_0) \right). \tag{3.51}$$

By again first making the necessary adjustment to the domain to ensure the differentiability of the integrand, we can integrate over the altered domain and use the divergence theorem to obtain the integral representation of u. By also making a switch of indices for a cleaner expression, we get

$$u_j(\boldsymbol{x}_0) = \frac{1}{\mu} \int_\Gamma G_{ij}(\boldsymbol{x}, \boldsymbol{x}_0) \sigma_{ik}(\boldsymbol{x}) n_k(\boldsymbol{x}) dl(\boldsymbol{x}) - \int_\Gamma u_i(\boldsymbol{x}) T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0) n_k(\boldsymbol{x}) dl(\boldsymbol{x}). \tag{3.52}$$

The first term of Equation (3.52) is the single layer potential, and its strength is given by the boundary traction, $\boldsymbol{\sigma} \cdot \boldsymbol{n}$. The second term is the double layer potential, and its strength is given by the boundary values.

3.5.2   Single layer integral representation

In a similar fashion as we saw in Section 3.2, we can define two solutions, one on the interior domain Ω and another on the region exterior to Γ. If we let these two solutions have equal values on boundary, we can take Equation (3.52) for $\boldsymbol{u}^{Int}$ and a similar equation for $\boldsymbol{u}^{Ext}$,

$$0 = -\frac{1}{\mu} \int_{\Gamma} G_{ij}(\boldsymbol{x}, \boldsymbol{x}_0) \sigma_{ik}^{Ext}(\boldsymbol{x}) n_k(\boldsymbol{x}) dl(\boldsymbol{x}) + \int_{\Gamma} u_i^{Ext}(\boldsymbol{x}) T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0) n_k(\boldsymbol{x}) dl(\boldsymbol{x}), \tag{3.53}$$

where $\boldsymbol{n}$ is pointing out of Ω. Then, by adding these equations, we get the single layer integral equation for u^{Int} given by

$$u_j^{Int}(\boldsymbol{x}_0) = \frac{1}{\mu} \int_{\Gamma} G_{ij}(\boldsymbol{x}, \boldsymbol{x}_0) \Phi_i(\boldsymbol{x}) dl(\boldsymbol{x}), \tag{3.54}$$

where the strength of the single layer potential is given by the jump in traction across the boundary, or

$$\boldsymbol{\Phi} = (\boldsymbol{\sigma}^{Int} - \boldsymbol{\sigma}^{Ext}) \cdot \boldsymbol{n} \Big|_{\Gamma}. \tag{3.55}$$

Force and Torque.   It will be useful for subsequent chapters to calculate the net force and net torque exerted on a closed curve. As we have established, the boundary distribution, $\boldsymbol{\Phi}$ is supported on the boundary Γ. Let $\widetilde{\Gamma}$ be another curve that encloses Γ. Then the net force on $\widetilde{\Gamma}$ is given by

$$\boldsymbol{B} = \int_{\widetilde{\Gamma}} \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}), \tag{3.56}$$

where we use $\boldsymbol{B}$ instead of $\boldsymbol{F}$ in order to avoid conflict with our use of $\boldsymbol{F}$ as the IBSL constraint force. Let R be the area enclosed by $\widetilde{\Gamma}$. Then we can use the divergence theorem to find

$$0 = \int_R \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x}) d\boldsymbol{x} = \int_{\widetilde{\Gamma}} \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) + \int_{\Gamma} [\![\boldsymbol{\sigma} \cdot \boldsymbol{n}]\!] dl(\boldsymbol{x}). \tag{3.57}$$

Therefore, the net force is

$$\boldsymbol{B} = -\int_{\Gamma} [\![\boldsymbol{\sigma} \cdot \boldsymbol{n}]\!] dl(\boldsymbol{x}). \tag{3.58}$$

In the case where $\boldsymbol{Phi}$ is defined as we have in this section, the net force is then

$$\boldsymbol{B} = -\int_{\Gamma} \boldsymbol{\Phi}(\boldsymbol{x}) dl(\boldsymbol{x}). \tag{3.59}$$

Then, the net torque on $\widetilde{\Gamma}$, in two dimensions, is given by

$$\boldsymbol{L} = \int_{\widetilde{\Gamma}} \boldsymbol{x} \times (\boldsymbol{\sigma} \cdot \boldsymbol{n}) dl(\boldsymbol{x}) = \epsilon_{ijk} \int_{\widetilde{\Gamma}} x_j \sigma_{km} n_m dl(\boldsymbol{x}), \tag{3.60}$$

where ϵ_{ijk} is the Levi-Civita symbol, which in 2-D gives 1 for an even permutation of $(1, 2)$, -1 for an odd permutation, and 0 otherwise. Before deriving the expression for the net torque, let us note that

$$\epsilon_{ijk} \int_{R} \frac{\partial}{\partial x_m} (x_j \sigma_{km}) d\boldsymbol{x} = \epsilon_{ijk} \int_{R} x_j \frac{\partial \sigma_{km}}{\partial x_m} d\boldsymbol{x} + \epsilon_{ijk} \int_{R} \delta_{mj} \sigma_{km} d\boldsymbol{x} = 0, \tag{3.61}$$

where in the penultimate expression, the first integral is 0 since $\nabla \cdot \boldsymbol{\sigma} = 0$ and the second integral is 0 by the symmetry of $\boldsymbol{\sigma}$. Then, we can use the divergence theorem to get

$$0 = \epsilon_{ijk} \int_{R} \frac{\partial}{\partial x_m} (x_j \sigma_{km}) d\boldsymbol{x} = \epsilon_{ijk} \int_{\widetilde{\Gamma}} x_j \sigma_{km} n_m dl(\boldsymbol{x}) + \epsilon_{ijk} \int_{\Gamma} x_j [\![\sigma_{km} n_m]\!] dl(\boldsymbol{x}). \tag{3.62}$$

Therefore the net torque exerted on $\widetilde{\Gamma}$ is

$$\boldsymbol{L} = -\epsilon_{ijk} \int_{\Gamma} x_j [\![\sigma_{km} n_m]\!] dl(\boldsymbol{x}). \tag{3.63}$$

Note that in two dimensions, this quantity can be viewed as a scalar. As with the net force, in terms of the single layer potential strength, this is equivalent to

$$L = -\int_{\Gamma} \boldsymbol{x} \times \boldsymbol{\Phi}(\boldsymbol{x}) dl(\boldsymbol{x}). \tag{3.64}$$

3.5.3 Double layer integral representation

If we follow the the same process as the previous section but require that the interior and exterior solutions have equal boundary tractions instead of equal boundary values, then we get the double layer integral equation for u^{Int} given by

$$u_j^{Int}(\boldsymbol{x}_0) = \int_\Gamma \Psi_i(\boldsymbol{x}) T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0) n_k(\boldsymbol{x}) dl(\boldsymbol{x}), \tag{3.65}$$

where the strength of the double layer potential is given by the jump in velocity across the boundary, or

$$\boldsymbol{\Psi} = (\boldsymbol{u}^{Ext} - \boldsymbol{u}^{Int})\Big|_\Gamma. \tag{3.66}$$

We now seek to find the net force and net torque on a closed curve $\widetilde{\Gamma}$ surrounding Γ. We can still use the derivations in Equations (3.57) and (3.62), but since the double layer representation was formed by assuming no jump in the traction across Γ, we get that the net force and net torque on $\widetilde{\Gamma}$ are 0. This shows us that the double layer potential alone is incapable of representing flows with a net force or net torque. For a domain interior to Γ, this does not cause a problem since a solution to Stokes equation on this domain has no net force, as seen by

$$\boldsymbol{B} = \int_\Gamma \boldsymbol{\sigma} \cdot \boldsymbol{n} dl(\boldsymbol{x}) = \int_\Omega \nabla \cdot \boldsymbol{\sigma} d\boldsymbol{x} = 0, \tag{3.67}$$

and similarly for the net torque. However, on an exterior domain, we require the ability to represent a flow with net force and net torque. In order to achieve this, we can supplement the double layer potential with a flow producing a finite force and torque on Γ. The resulting representation is called a compound or completed double layer representation. For a review of ways to do this, one can see Section 4.7 in [50]. We briefly present two such methods here.

One solution, presented by Power and Miranda [48, 49], is to add a pair of singularities to the interior of Ω using a Stokeslet, or a point force, and a rotlet, or a point torque. A second solution, which we will make explicit use of in Chapter 6, was presented by Hsiao and Kress [34] and Hebeker [33]. It consists of adding a flow created by a single layer potential whose strength is a constant multiple of the double layer potential strength. If we let this constant be η/μ, where η

is an arbitrary positive constant, then our integral representation is given by

$$u_j(\boldsymbol{x}_0) = \int_\Gamma \Psi_i(\boldsymbol{x})T_{ijk}(\boldsymbol{x},\boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) + \frac{\eta}{\mu}\int_\Gamma G_{ij}(\boldsymbol{x},\boldsymbol{x}_0)\Psi_i(\boldsymbol{x})dl(\boldsymbol{x}). \tag{3.68}$$

Using what we found in Section 3.5.2, we then have the net force and net torque as

$$\boldsymbol{B} = -\int_\Gamma \boldsymbol{\Psi}(\boldsymbol{x})dl(\boldsymbol{x}) \tag{3.69a}$$

$$L = -\int_\Gamma \boldsymbol{x} \times \boldsymbol{\Psi}(\boldsymbol{x})dl(\boldsymbol{x}). \tag{3.69b}$$

3.6 Integral equations for Stokes equation with Dirichlet boundary conditions

In this section, we introduce the integral equations that arise from using the representations from the previous section to solve the Stokes equation with Dirichlet boundary conditions given by

$$\mu\Delta\boldsymbol{u} - \nabla p = \nabla \cdot \boldsymbol{\sigma} = 0 \qquad \text{in } \Omega \tag{3.70a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{3.70b}$$

$$\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma. \tag{3.70c}$$

We omit the derivation, as it is similar to that in Section 3.3, but using the property of $\boldsymbol{T}$ given in Equation (3.43), we get the single and double layer integral equations given by

$$U_{b_j}(\boldsymbol{X}_0) = \frac{1}{\mu}\int_\Gamma G_{ij}(\boldsymbol{x},\boldsymbol{X}_0)\Phi_i(\boldsymbol{x})dl(\boldsymbol{x}), \tag{3.71}$$

$$U_{b_j}(\boldsymbol{X}_0) = \int_\Gamma^{PV} \Psi_i(\boldsymbol{x})T_{ijk}(\boldsymbol{x},\boldsymbol{X}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) - \frac{1}{2}\Psi_j(\boldsymbol{X}_0), \tag{3.72}$$

where the unknown quantities in Equations (3.71) and (3.72) are the potential strengths, $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, respectively. Note that again, the integral equations for the Brinkman equation have the same form. For the reasons discussed in Section 3.5.3, the double layer integral equation above is only valid for flows with no net force or torque. For other exterior flows, we can use the completed

41

double layer integral equation given by

$$U_{b_j}(\boldsymbol{X}_0) = \int_\Gamma \Psi_i(\boldsymbol{x})T_{ijk}(\boldsymbol{x},\boldsymbol{X}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) + \frac{\eta}{\mu}\int_\Gamma G_{ij}(\boldsymbol{x},\boldsymbol{X}_0)\Psi_i(\boldsymbol{x})dl(\boldsymbol{x}) - \frac{1}{2}\Psi_j(\boldsymbol{X}_0),$$

(3.73)

which comes from taking the limit as $\boldsymbol{x}_0 \longrightarrow \Gamma$ in Equation (3.68).

3.7 Discussion

Boundary integral methods can be used to get very accurate solutions to homogeneous linear boundary value problems by reformulating them as integral equations with one fewer dimension. However, the efficiency is greatly reduced when the problem is nonhomogeneous as this introduces an integral over the entire PDE domain. It can also be expensive to use the potential strength to obtain solution values at all points in a discretized domain. Additionally, by requiring the analytical form of the Green's function and its derivative, it can be difficult to generalize to non-free-space exterior domains, such as the periodic box used in this dissertation. Therefore, as discussed in Section 2.5, we seek to use the Immersed Boundary framework instead of explicitly using boundary integral methods. However, boundary integral methods have the advantage that well-conditioned integral operators exist, and this makes the step of solving for the unknown boundary distribution much more efficient.

Let us examine the integral operators for Laplace's equation to explore this conditioning issue. The single layer integral equation,

$$U_b(\boldsymbol{X}_0) = \int_\Gamma \sigma(\boldsymbol{x})G(\boldsymbol{x},\boldsymbol{X}_0)dl(\boldsymbol{x}),$$

(3.74)

has the form of a Fredholm integral of the first kind, and the double layer integral equation,

$$U_b(\boldsymbol{X}_0) = \int_\Gamma^{PV} \gamma(\boldsymbol{x})\nabla G(\boldsymbol{x},\boldsymbol{X}_0)\cdot\boldsymbol{n}(\boldsymbol{x})dl(\boldsymbol{x}) - \frac{1}{2}\gamma(\boldsymbol{X}_0),$$

(3.75)

has the form of a Fredholm integral of the second kind [3]. The operators given by

$$K_1\sigma = \int_\Gamma \sigma(\boldsymbol{x})G(\boldsymbol{x},\boldsymbol{x}_0)dl(\boldsymbol{x}) \tag{3.76a}$$

$$K_2\gamma = \int_\Gamma \gamma(\boldsymbol{x})\nabla G(\boldsymbol{x},\boldsymbol{x}_0)\cdot \boldsymbol{n}(\boldsymbol{x})dl(\boldsymbol{x}) \tag{3.76b}$$

have eigenvalues in the interval $(-0.5, 0.5)$, and the only limit point of the eigenvalues is 0 [1, 13]. The condition number, defined as the ratio of the largest and smallest eigenvalues, is therefore infinite in the continuous case and large in the discretized case. On the other hand, Equation (3.75) can be rewritten as

$$\left(K_2 - \frac{1}{2}\mathbb{I}\right)\gamma = U_b, \tag{3.77}$$

where $\mathbb{I}$ is the identity operator. Shifting the operator in this way shifts the eigenvalues, and the only limit point for the eigenvalues becomes -1/2. Therefore, the condition number of the discretized operator is finite and does not grow with refinement of the discretization. The better conditioning of this operator is the characteristic of the double layer representation that we exploit to form our new Immersed Boundary Double Layer method.

Chapter 4

Connection between IBSL method and boundary integral methods

In this chapter, we relate the Immersed Boundary Single Layer method as described in Chapter 2 to the single layer integral equations described in Chapter 3. As discussed in Chapter 1, this connection has been established in several recent works [**17**, **62**], but here, we present this connection explicitly from the IBSL constraint systems seen in Equation (2.5) for Helmholtz and Poisson equations and in Equation (2.37) for Brinkman and Stokes equations. By establishing this connection, we can see that the large iteration counts seen in Chapter 2 can be explained by the poor conditioning of the integral operator, as discussed at the end of Chapter 3. We will then propose that a solution to the conditioning problem is to formulate an Immersed Boundary version of a double layer integral equation. Such formulations will be presented in Chapters 5 and 6.

For this chapter and the derivation of the IBDL method, we will simplify the presentation by focusing on the homogeneous case, or $g = 0$. However, this connection holds for the inhomogeneous case as well, and we use the new method on several inhomogenous problems in Chapters 5 and 6.

4.1   Poisson and Helmholtz equations

We begin by making this connection explicit in the case of the Poisson and Helmholtz equations. For simplicity, we assume that the differential operator $\mathcal{L} = \Delta - k^2$ is invertible, either by focusing on Helmholtz or assuming a computational domain $\mathcal{C}$ for which Δ is invertible.

4.1.1 Regularized Green's function

Let us first define a regularized Green's function by

$$G_h(\boldsymbol{x}, \boldsymbol{x}_0) \equiv G * \delta_h = \int_{\mathcal{C}} G(\mathbf{y}, \boldsymbol{x}_0) \delta_h(\boldsymbol{x} - \mathbf{y}) d\mathbf{y}. \tag{4.1}$$

The linearity of the differential operator then gives us that

$$\mathcal{L}G_h(\boldsymbol{x}, \boldsymbol{x}_0) = -\delta_h(\boldsymbol{x} - \boldsymbol{x}_0). \tag{4.2}$$

In this way, G_h can be seen both as a regularization of a Green's function through a convolution with a smoothing function and as the solution to the PDE when the forcing function is a regularized delta function.

4.1.2 IBSL method as a regularized single layer integral equation

The homogeneous Immersed Boundary Single Layer system is given by

$$\mathcal{L}u + SF = 0 \qquad \text{in } \mathcal{C} \tag{4.3a}$$

$$S^*u = U_b \qquad \text{on } \Gamma. \tag{4.3b}$$

Starting with Equation (4.3a) and using the definition of the spread operator, we have

$$\mathcal{L}u = -SF = -\int_{\Gamma} F(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{4.4}$$

Inverting the operator and using Equation (4.2), we get

$$u(\boldsymbol{x}) = \int_{\Gamma} F(s)G_h(\boldsymbol{x}, \boldsymbol{X}(s))ds \tag{4.5}$$

for $\boldsymbol{x} \in \Omega \setminus \Gamma$. Then, Equation (4.3b) gives us

$$U_b(s') = \int_{\mathcal{C}} u(\boldsymbol{x})\delta_h(\boldsymbol{x} - \boldsymbol{X}(s'))d\boldsymbol{x}, \tag{4.6}$$

where we are using s' to distinguish from our previous variable of integration or, alternatively, viewing $\boldsymbol{X}(s')$ as selecting a particular boundary point. Combining this with Equation (4.5) and changing the order of integration, we get

$$U_b(s') = \int_\Gamma F(s) \int_\mathcal{C} G_h(\boldsymbol{x}, \boldsymbol{X}(s))\delta_h(\boldsymbol{x} - \boldsymbol{X}(s'))d\boldsymbol{x}ds. \tag{4.7}$$

Notice that the second integral is equivalent to $G_h * \delta_h$, evaluated at $\boldsymbol{X}(s')$. Therefore, by denoting a twice-regularized Green's function by G_{hh}, this becomes

$$U_b(s') = \int_\Gamma F(s)G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s))ds. \tag{4.8}$$

The symmetry of the Green's function, which is preserved through convolutions with the regularized delta function, gives us that $G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) = G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))$. Using this to switch the arguments of G_{hh} and appropriately redefining F and U_b as functions of $\boldsymbol{x}$, we get

$$U_b(\boldsymbol{X}(s')) = \int_\Gamma F(\boldsymbol{X}(s))G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))ds. \tag{4.9}$$

Recall from Equation (3.30) that the single layer integral equation for $\boldsymbol{X}_0$ on the boundary is given by

$$U_b(\boldsymbol{X}_0) = \int_\Gamma \sigma(\boldsymbol{x})G(\boldsymbol{x}, \boldsymbol{X}_0)dl(\boldsymbol{x}). \tag{4.10}$$

By parametrizing Γ by $\boldsymbol{X}(s)$, this becomes

$$U_b(\boldsymbol{X}(s')) = \int_\Gamma \sigma(\boldsymbol{X}(s))G(\boldsymbol{X}(s), \boldsymbol{X}(s'))\left|\frac{\partial \boldsymbol{X}(s)}{\partial s}\right|ds. \tag{4.11}$$

By comparing Equations (4.9) and (4.11), we can associate the IBSL constraint force F with the potential strength modified by the parametrization term, $\left|\partial \boldsymbol{X}(s)/\partial s\right|$. In other words, in the limit that the regularization width, h, approaches 0, we have

$$F(\boldsymbol{X}) = \sigma(\boldsymbol{X})\left|\frac{\partial \boldsymbol{X}}{\partial s}\right|. \tag{4.12}$$

Specifically, in the case of an arclength parametrization, where $\left|\partial \boldsymbol{X}(s)/\partial s\right| = 1$, we see from Chapter 3 that F gives the jump in the normal derivative of the solution across the boundary.

We have now established that the IBSL method is equivalent to a *regularized* single layer integral equation. It is this connection that motivates the name Immersed Boundary Single Layer method in order to distinguish it from the new method we propose in Chapter 5.

4.2   Stokes equation

We next look at establishing this connection between the IBSL method and a single layer integral equation for the Stokes equation. Again, for simplicity we assume that the differential operators involved are invertible on the computational domain $\mathcal{C}$. Section 2.3.3 discusses how to utilize the IBSL method when they are not invertible.

4.2.1   Regularized Green's functions and related identities

Since Stokes equation involves the Laplacian, we will require the use of the regularized Green's function for Laplace's equation developed in Section 4.1.1. To distinguish it from the Stokes Green's function in this section, we will call this function G_h^L. In this section, we will define other regularized Green's functions and derive some identities that will be used in Section 4.2.2 and in Chapter 6.

Regularized biharmonic Green's function.   Let us define the Green's function to the biharmonic equation as the function $H(\boldsymbol{x}, \boldsymbol{x}_0)$ that satisfies

$$\Delta^2 H = -\delta(\boldsymbol{x} - \boldsymbol{x}_0). \tag{4.13}$$

We also define the regularized biharmonic Green's function by

$$H_h(\boldsymbol{x}, \boldsymbol{x}_0) \equiv H * \delta_h = \int_{\mathcal{C}} H(\mathbf{y}, \boldsymbol{x}_0)\delta_h(\boldsymbol{x} - \mathbf{y})d\mathbf{y}. \tag{4.14}$$

The linearity of the differential operator again gives us that

$$\Delta^2 H_h(\boldsymbol{x}, \boldsymbol{x}_0) = -\delta_h(\boldsymbol{x} - \boldsymbol{x}_0). \tag{4.15}$$

Then using our regularized Laplace's Green's function, we have

$$\Delta^2 H_h(\boldsymbol{x}, \boldsymbol{x}_0) = \Delta G_h^L(\boldsymbol{x}, \boldsymbol{x}_0), \tag{4.16}$$

and therefore that

$$\Delta H_h(\boldsymbol{x}, \boldsymbol{x}_0) = G_h^L(\boldsymbol{x}, \boldsymbol{x}_0). \tag{4.17}$$

Note that we also have this relationship for the exact Green's functions, so

$$\Delta H(\boldsymbol{x}, \boldsymbol{x}_0) = G^L(\boldsymbol{x}, \boldsymbol{x}_0). \tag{4.18}$$

Relationship between Stokes and biharmonic Green's functions. We next seek to find a relationship between the Stokes Green's function $\boldsymbol{G}$ and the biharmonic Green's function H. We start with the singularly forced Stokes equation, given by

$$\mu \Delta \boldsymbol{u} - \nabla p = -\delta(\boldsymbol{x} - \boldsymbol{x}_0)\boldsymbol{c} \tag{4.19a}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{4.19b}$$

By taking the divergence of Equation (4.19a) and using the incompressibility of the velocity, we get

$$\Delta p = \nabla \cdot \delta(\boldsymbol{x} - \boldsymbol{x}_0)\boldsymbol{c}. \tag{4.20}$$

Then we can invert the operator to get

$$p(\boldsymbol{x}) = -\nabla \cdot G^L(\boldsymbol{x}, \boldsymbol{x}_0)\boldsymbol{c} = -\nabla G^L(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{c}. \tag{4.21}$$

Using this in Equation (4.19a), we get

$$\mu \Delta \boldsymbol{u} = -\nabla(\nabla G^L(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{c}) - \delta(\boldsymbol{x} - \boldsymbol{x}_0)\boldsymbol{c}. \tag{4.22}$$

Again inverting the operator and using Equation (4.18) for the first term and Equation (4.13) for the second term, we get

$$\mu\boldsymbol{u}(\boldsymbol{x}) = -\nabla(\nabla H(\boldsymbol{x},\boldsymbol{x}_0)\cdot\boldsymbol{c}) + \Delta H(\boldsymbol{x},\boldsymbol{x}_0)\boldsymbol{c}. \tag{4.23}$$

Switching to Einstein notation, we have

$$\mu u_i(\boldsymbol{x}) = -\frac{\partial H}{\partial x_i \partial x_j}(\boldsymbol{x},\boldsymbol{x}_0)c_j + \frac{\partial H}{\partial x_k \partial x_k}(\boldsymbol{x},\boldsymbol{x}_0)c_i. \tag{4.24}$$

Therefore, since the the solution using the Stokes Green's function is given by

$$u_i(\boldsymbol{x}) = \frac{1}{\mu}G_{ij}(\boldsymbol{x},\boldsymbol{x}_0)c_j, \tag{4.25}$$

we see that

$$G_{ij} = -\frac{\partial H}{\partial x_i \partial x_j} + \delta_{ij}\frac{\partial H}{\partial x_k \partial x_k}. \tag{4.26}$$

Relationship between the Stokes pressure vector and Laplace's Green's function.
We will now find an identity for the Stokes pressure vector that will be needed in Section 6.1. From equation (4.21), we have

$$p(\boldsymbol{x}) = -\nabla G^L(\boldsymbol{x},\boldsymbol{x}_0)\cdot\boldsymbol{c}. \tag{4.27}$$

Since the pressure vector is defined in Equation (3.34b) as $\boldsymbol{p}$ such that $p(\boldsymbol{x}) = p_j c_j$, we can see that

$$p_j = -\frac{\partial G^L}{\partial x_j}(\boldsymbol{x},\boldsymbol{x}_0). \tag{4.28}$$

Regularized Stokes Green's function.　Let us also define the regularized Stokes Green's function and the corresponding regularized pressure vector and stress tensor as

$$G_{ij}^h(\boldsymbol{x},\boldsymbol{x}_0) \equiv G_{ij} * \delta_h = \int_{\mathcal{C}} G_{ij}(\mathbf{y},\boldsymbol{x}_0)\delta_h(\boldsymbol{x}-\mathbf{y})d\mathbf{y}, \tag{4.29a}$$

$$p_j^h(\boldsymbol{x}, \boldsymbol{x}_0) \equiv p_j * \delta_h = \int_{\mathcal{C}} p_j(\mathbf{y}, \boldsymbol{x}_0)\delta_h(\boldsymbol{x} - \mathbf{y})d\mathbf{y}, \tag{4.29b}$$

$$T_{ijk}^h(\boldsymbol{x}, \boldsymbol{x}_0) \equiv T_{ijk} * \delta_h = \int_{\mathcal{C}} T_{ijk}(\mathbf{y}, \boldsymbol{x}_0)\delta_h(\boldsymbol{x} - \mathbf{y})d\mathbf{y}, \tag{4.29c}$$

By linearity, the relationships that we have previously shown for the exact functions also hold for the regularized ones, so that

$$G_{ij}^h = -\frac{\partial H_h}{\partial x_i \partial x_j} + \delta_{ij}\frac{\partial H_h}{\partial x_k \partial x_k}, \tag{4.30}$$

and the solution to a Stokes equation that is forced with the regularized delta function will exactly be the solution corresponding to this regularized Green's function. Additionally, we have

$$\boldsymbol{p}_j^h = -\frac{\partial G_h^L}{\partial x_j}(\boldsymbol{x}, \boldsymbol{x}_0). \tag{4.31}$$

4.2.2 IBSL method as a regularized single layer integral equation

The homogeneous Immersed Boundary Single Layer system for Stokes equation is given by

$$\mu\Delta\boldsymbol{u} - \nabla p + S\boldsymbol{F} = 0 \qquad \text{in } \mathcal{C} \tag{4.32a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \mathcal{C} \tag{4.32b}$$

$$S^*\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma. \tag{4.32c}$$

We will now explicitly derive the connection to a regularized single layer integral equation. Many of the steps will mimic the steps taken in Section 4.2.1. Taking the divergence of Equation (4.32a) and using Equation (4.32b) to eliminate the velocity term, we get

$$\Delta p = \nabla \cdot S\boldsymbol{F} = \nabla \cdot \int_{\Gamma} \boldsymbol{F}(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{4.33}$$

We can then invert the operator and use the property of the regularized Green's function for Laplace's equation, given in Equation (4.2). We then have

$$p(\boldsymbol{x}) = -\nabla \cdot \int_\Gamma \boldsymbol{F}(s) G_h^L(\boldsymbol{x}, \boldsymbol{X}(s)) ds. \tag{4.34}$$

Bringing in the divergence and manipulating the expression, we get

$$p(\boldsymbol{x}) = -\int_\Gamma \boldsymbol{F}(s) \cdot \nabla G_h^L(\boldsymbol{x}, \boldsymbol{X}(s)) ds. \tag{4.35}$$

Using this expression for pressure, Equation (4.32a) then gives us

$$\mu \Delta \boldsymbol{u} = \nabla p - S\boldsymbol{F} = -\nabla \int_\Gamma \boldsymbol{F}(s) \cdot \nabla G_h^L(\boldsymbol{x}, \boldsymbol{X}(s)) ds - \int_\Gamma \boldsymbol{F}(s) \delta_h(\boldsymbol{x} - \boldsymbol{X}(s)) ds. \tag{4.36}$$

At this point, we switch to Einstein summation notation, and by bringing in the gradient, we get

$$\mu \Delta u_j = -\int_\Gamma \frac{\partial^2 G_h^L}{\partial x_i \partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) F_i(s) ds - \int_\Gamma F_j(s) \delta_h(\boldsymbol{x} - \boldsymbol{X}(s)) ds. \tag{4.37}$$

Inverting the operator on u_i and using Equation (4.17) for the first term and Equation (4.15) for the second term, we get

$$\mu u_j(\boldsymbol{x}) = -\int_\Gamma \frac{\partial^2 H_h}{\partial x_i \partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) F_i(s) ds + \int_\Gamma F_j(s) \frac{\partial^2 H_h}{\partial x_k \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s)) ds. \tag{4.38}$$

Recognizing the form of the regularized Stokes Green's function from Equation (4.30), we have

$$u_j(\boldsymbol{x}) = \frac{1}{\mu} \int_\Gamma G_{ij}^h(\boldsymbol{x}, \boldsymbol{X}(s)) F_i(s) ds \tag{4.39}$$

for $\boldsymbol{x} \in \Omega \setminus \Gamma$. Next, we use Equation (4.32c) to get

$$\boldsymbol{U}_b(s') = \int_\mathcal{C} \boldsymbol{u}(\boldsymbol{x}) \delta_h(\boldsymbol{x} - \boldsymbol{X}(s')) dx. \tag{4.40}$$

We then follow the same steps seen in Section 4.1.2. We combine this with Equation (4.38), change the order of integration, recognize the presence of $G_{ij}^h * \delta_h$, and denote the twice-regularized Green's function by G_{ij}^{hh}. This leaves us with

$$U_{b_j}(s') = \frac{1}{\mu} \int_\Gamma G_{ij}^{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) F_i(s) ds. \tag{4.41}$$

The symmetry of the Green's function, which is preserved through convolutions with the regularized delta function, gives us that $G_{ij}^{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) = G_{ij}^{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))$ [50]. Using this to switch the arguments of G_{hh} and appropriately redefining $\boldsymbol{F}$ and $\boldsymbol{U}_b$ as functions of $\boldsymbol{x}$, we get

$$U_{b_j}(\boldsymbol{X}(s')) = \frac{1}{\mu} \int_\Gamma G_{ij}^{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))F_i(s)ds. \tag{4.42}$$

Recall that the single layer integral equation for $\boldsymbol{X}_0$ on the boundary from Section 3.6 is given by

$$U_{b_j}(\boldsymbol{X}_0) = \frac{1}{\mu} \int_\Gamma G_{ij}(\boldsymbol{x}, \boldsymbol{X}_0)\Phi_i(\boldsymbol{x})dl(\boldsymbol{x}). \tag{4.43}$$

By parametrizing Γ by $\boldsymbol{X}(s)$, we get

$$U_{b_j}(\boldsymbol{X}(s')) = \frac{1}{\mu} \int_\Gamma G_{ij}(\boldsymbol{X}(s), \boldsymbol{X}(s'))\Phi_i(\boldsymbol{X}(s))\left|\frac{\partial \boldsymbol{X}(s)}{\partial s}\right|ds. \tag{4.44}$$

By comparing Equations (4.42) and (4.44), in the case of an arclength parametrization, we can associate the IBSL constraint force $\boldsymbol{F}$ with the potential strength $\boldsymbol{\Phi}$, and in the limit that the regularization width, h, approaches 0, we have

$$\boldsymbol{F}(\boldsymbol{X}) = \Phi(\boldsymbol{X}). \tag{4.45}$$

We then see from Chapter 3 that $\boldsymbol{F}$ gives the jump in the traction across the boundary. We have now established that the IBSL method for Stokes is equivalent to a *regularized* single layer integral equation.

Chapter 5

The Immersed Boundary Double Layer (IBDL) method: Helmholtz and Poisson equations

In this chapter, we introduce the Immersed Boundary Double Layer (IBDL) method for Helmholtz and Poisson equations. The development of this method was motivated by the desire to maintain the robustness of the Immersed Boundary method for boundary value problems, while avoiding the poor conditioning introduced by the form of the IBSL method, namely the first-kind integral equation to which it corresponds. In this chapter, we see that the IBDL method is in fact able to achieve the same first-order convergence as the IBSL method while improving this conditioning drastically by utilizing the form of a second-kind integral equation. In this chapter, we also explore other advantages of the IBDL method, such as the convergence of the potential strength. This convergence also allows us to use the framework of the IBDL method to solve a PDE with Neumann boundary conditions. This is in contrast to the IBSL method, which only applies to Dirichlet problems.

This chapter is organized as follows. In Section 5.1, we introduce the method and demonstrate the connection to a regularized double layer integral equation. We also discuss the discontinuity of the resulting solution and how this affects pointwise convergence near the boundary. In Section 5.2, we present the method for using the IBDL method to solve PDEs with Neumann boundary conditions. In Section 5.3, we discuss the numerical implementation, focusing on the areas in which this is different from the IBSL numerical implementation discussed in Section 2.2. In this section, we also discuss the way in which the IBDL method can be used to easily flag points as interior or exterior to the immersed boundary. In Section 5.4, we compute solutions to Helmholtz and Poisson equations and compare the results to the IBSL method. In Section 5.5, we further discuss the factors affecting the pointwise convergence of the solution. In Section 5.6, we illustrate the

convergence of the strength of the potential. Finally, in Section 5.7, we solve a PDE with Neumann boundary conditions.

5.1 Mathematical description of the method

We look again at the boundary value problem given by

$$\mathcal{L}u = g \qquad \text{in } \Omega \tag{5.1a}$$

$$u = U_b \qquad \text{on } \Gamma, \tag{5.1b}$$

where $\mathcal{L} = \Delta - k^2$ is the Helmholtz or Laplacian operator. As in Chapter 2, we assume that Ω is a subset of the larger computational domain $\mathcal{C}$ and that Γ is a smooth one-dimensional curve parametrized by $\boldsymbol{X}(s)$.

5.1.1 Formulation

The Immersed Boundary Double Layer formulation for Equation (5.1) is given by

$$\mathcal{L}u + \widetilde{S}Q = \tilde{g} \qquad \text{in } \mathcal{C} \tag{5.2a}$$

$$S^* u + \frac{1}{2}Q = U_b \qquad \text{on } \Gamma, \tag{5.2b}$$

where we assume an arclength parametrization of Γ, and where we define the operator $\widetilde{S}$ as

$$\widetilde{S}Q \equiv \nabla \cdot (SQ\boldsymbol{n}), \tag{5.3}$$

for $\boldsymbol{n}$, the unit normal pointing out of Ω. Using the definition of S, this gives us

$$(\widetilde{S}Q)(\boldsymbol{x}) = \nabla \cdot \int_{\Gamma} Q(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds = \int_{\Gamma} Q(s)\nabla \cdot \delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{5.4}$$

The form of $\widetilde{S}$ then implies that Q gives the strength of a dipole force distribution on the boundary, and this is our Lagrange multiplier used to enforce the boundary condition. Recall that $\tilde{g}$ represents an extension of the function g from Ω to the computational domain $\mathcal{C}$, and we define it as $\tilde{g} \equiv$

$g\chi_\Omega + g_e\chi_{C\setminus\Omega}$. For the Helmholtz equation, there is flexibility in our choice for g_e. However, we will see in Section 5.3.3 that we can sometimes use our choice of g_e to deal with the nullspace of the periodic Laplacian.

We will next demonstrate that this formulation corresponds to a regularized double layer integral equation. For simplicity, we will consider the homogeneous case, $g = 0$. Starting with Equation (5.2a), we have

$$\mathcal{L}u = -\widetilde{S}Q = -\nabla \cdot \int_\Gamma Q(s)\boldsymbol{n}(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{5.5}$$

Inverting the operator and using the property of the regularized Green's function given by Equation (4.2), we get

$$u(\boldsymbol{x}) = \nabla \cdot \int_\Gamma Q(s)\boldsymbol{n}(s)G_h(\boldsymbol{x}, \boldsymbol{X}(s))ds \tag{5.6}$$

for $\boldsymbol{x} \in \Omega \setminus \Gamma$. Bringing in the divergence and manipulating the expression, we get

$$u(\boldsymbol{x}) = \int_\Gamma Q(s)\nabla G_h(\boldsymbol{x}, \boldsymbol{X}(s)) \cdot \boldsymbol{n}(s)ds. \tag{5.7}$$

Then, the second equation of the IBDL method, Equation (5.2b) gives us

$$U_b(s') = \int_\mathcal{C} u(\boldsymbol{x})\delta_h(\boldsymbol{x} - \boldsymbol{X}(s'))dx + \frac{1}{2}Q(s'). \tag{5.8}$$

Combining Equation (5.7) with Equation (5.8), changing the order of integration, and again recognizing the presence of $G_{hh} = G_h * \delta_h$, we get

$$U_b(s') = \int_\Gamma Q(s)\nabla G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) \cdot \boldsymbol{n}(s)ds + \frac{1}{2}Q(s'). \tag{5.9}$$

The odd symmetry of the gradient of the Green's function [51], which is again preserved through the convolutions with the regularized delta function, gives us that $\nabla G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) = -\nabla G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))$. Using this to switch the arguments of ∇G_{hh} and appropriately redefining Q, U_b, and $\boldsymbol{n}$ as functions of $\boldsymbol{x}$, we get

$$U_b(\boldsymbol{X}(s')) = -\int_\Gamma Q(\boldsymbol{X}(s))\nabla G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s')) \cdot \boldsymbol{n}(\boldsymbol{X}(s))ds + \frac{1}{2}Q(\boldsymbol{X}(s')). \tag{5.10}$$

Recall that the double layer integral equation for $\boldsymbol{X}_0$ on the boundary is given by

$$U_b(\boldsymbol{X}_0) = \int_\Gamma^{PV} \gamma(\boldsymbol{x})\nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x})dl(\boldsymbol{x}) - \frac{1}{2}\gamma(\boldsymbol{X}_0). \tag{5.11}$$

Using our arclength parametrization, we can rewrite this as

$$U_b(\boldsymbol{X}(s')) = \int_\Gamma^{PV} \gamma(\boldsymbol{X}(s))\nabla G(\boldsymbol{X}(s), \boldsymbol{X}(s')) \cdot \boldsymbol{n}(\boldsymbol{X}(s))ds - \frac{1}{2}\gamma(\boldsymbol{X}(s')). \tag{5.12}$$

By comparing Equations (5.10) and (5.12), we see that in the limit that the regularization width, h, approaches 0, we have

$$Q(\boldsymbol{X}) = -\gamma(\boldsymbol{X}). \tag{5.13}$$

Therefore, we can see that the IBDL method is equivalent to a *regularized* double layer integral equation. Note from Section 3.3 that Q gives the jump in the solution across the boundary.

We reiterate that the formulation, specifically Equation (5.2b), assumes that $X(s)$ gives an arclength parametrization of Γ. In the IBSL method, if one uses a different paramterization of the boundary, the rescaling of F seen in Equation (4.12) accounts for the transformation, and no alteration in the method is needed to find the solution. However, in the IBDL method, since the $1/2$ in Equation (5.2b) is derived using an arclength parametrization, one would need to alter this term in the case that $|\partial\boldsymbol{X}(s)/\partial s| \neq 1$. This will be discussed further in Section 5.1.3, but otherwise, we will assume an arclength parametrization throughout this dissertation.

5.1.2 Discontinuity of the solution

Since our boundary density Q corresponds to the jump in the solution values across the boundary, clearly our solution u will be discontinuous. In the original IBSL method, the solution was continuous, but the normal derivative was not. Figure 5.1 illustrates the solutions to Equation (2.1) produced by the IBSL and IBDL methods on the entire computational domain, $\mathcal{C} = [-0.5, 0.5]^2$. The PDE domain Ω is the interior of a circle, and the boundary condition is $U_b = e^x$. We can see that the two solutions match on the portion of Ω that is about a couple meshwidths away from the boundary. However, the IBDL method gives a solution that is discontinuous across the boundary.
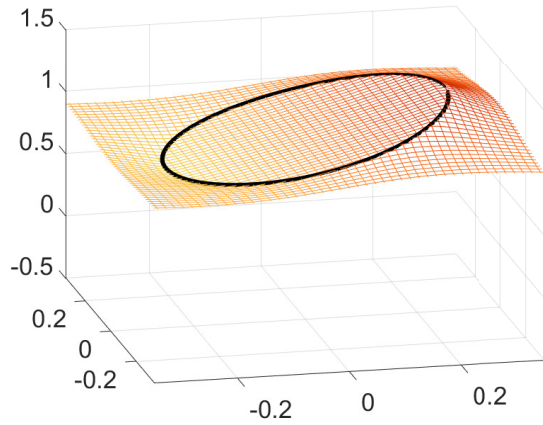
(a) IBSL Solution
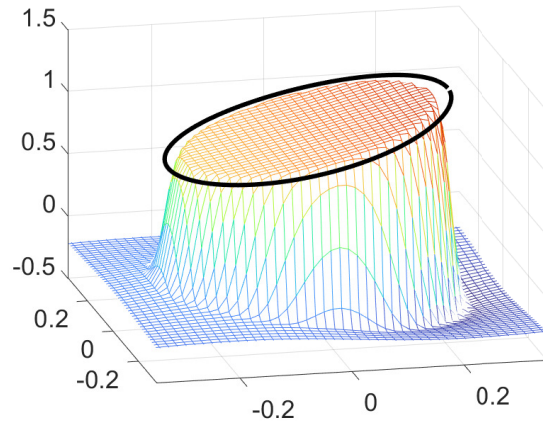
(b) IBDL Solution
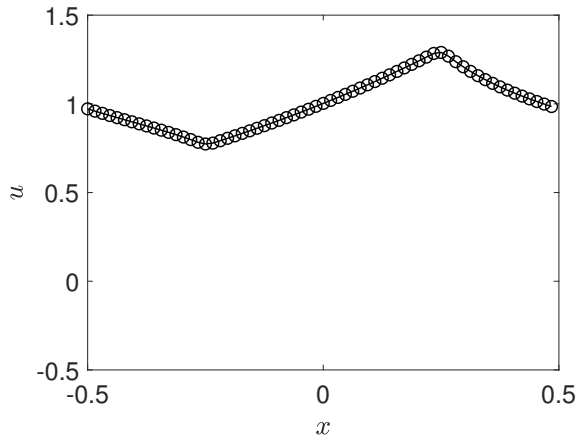
(c) IBSL Solution
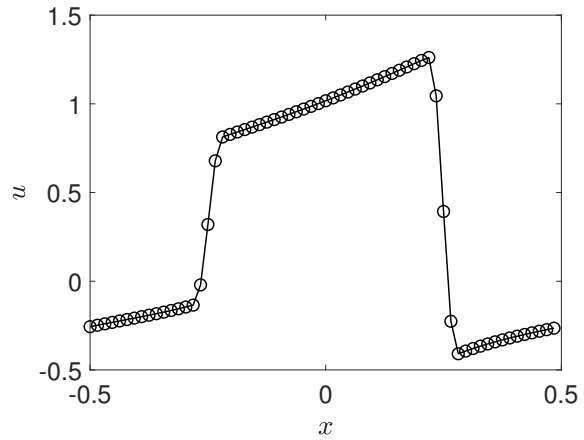
(d) IBDL Solution

FIGURE 5.1. Solution plots found using the IBSL and IBDL methods to solve Equation (5.1) with $k = 1$ and $g = 0$, and where the circular boundary of radius 0.25 has prescribed boundary values given by $U_b = e^x$. The periodic computational domain is $[-0.5, 0.5]^2$, and the grid and boundary point spacing is $\Delta s \approx \Delta x = 2^{-6}$. Figure 5.1a gives the solution from using the IBSL method, and Figure 5.1b, the IBDL method. U_b is shown with a black curve. Figures 5.1c and 5.1d show solution values for $y = -0.015625$ for the IBSL and IBDL methods, respectively.

The lack of smoothness in the derivative across the boundary causes the IBSL method to achieve only first-order accuracy. Since the IBDL method will instead yield a discontinuous function, we will not see pointwise convergence near the boundary if we use the solution values given directly from the method in Equation (5.2). We therefore must replace the values of u for grid points *near* the boundary.

Integral equation methods are generally able to achieve higher accuracy than the Immersed Boundary method. However, this lack of smoothness related to the singularity in the Green's function derivative results in the need to employ analytical techniques in order to achieve the same level of accuracy for points near Γ. For instance, Beale and Lai use a regularized Green's function and then analytically derive the correction terms [5]. Klöckner et al. use analytical expansions centered at points several meshwidths from the boundary to evaluate the solution for points nearer to the boundary [37], and Carvalho et al. use asymptotic analysis to approximate the solution at near-boundary points with known boundary data and a nonlocal correction [11].

In the Immersed Boundary method, however, one only expects first-order accuracy, and we are able to obtain this level of accuracy with a simple linear interpolation using known boundary values and approximate solution values several meshwidths into the PDE domain. Section 5.3.4 provides details of the numerical implementation.

There are two regions of grid points for which we do not see pointwise convergence. The first region contains the largest errors, and it is illustrated in Figure 5.2 for an interior circular domain with boundary data $U_b = e^x$. We can see that this region remains localized to only a small number of grid points, so that as we refine the grid, the width of this region quickly goes to 0. These large errors are the direct result of the smoothing of the dipole forces, and the number of meshwidths is therefore determined by the support of the regularized delta function used. As discussed in Section 2.2.2, we generally use a 4-point delta function, and this results in about $2 - 3$ meshdwidths of large errors on one side of the boundary, which can be seen in Figure 5.2.

Secondly, the numerical method used to discretize the PDE spreads the error from the discontinuity into a larger region near the boundary. Therefore, if we use the interpolation to only correct for the large errors within the first few meshwidths, we will still see a lessening of first-order convergence in the max norm for sufficiently fine grids. The width of the region on which the solution fails to converge pointwise still approaches 0, but the number of meshwidths affected can increase. One can recover pointwise convergence with further interpolation, and in practice, one can generally obtain a maximum error comparable to that of the IBSL method by using interpolation for only a relatively small number of meshwidths. In Section 5.5, we further discuss factors that contribute to the number of meshwidths needed.

58

(a) $\Delta x = 2^{-8}$ (b) $\Delta x = 2^{-12}$
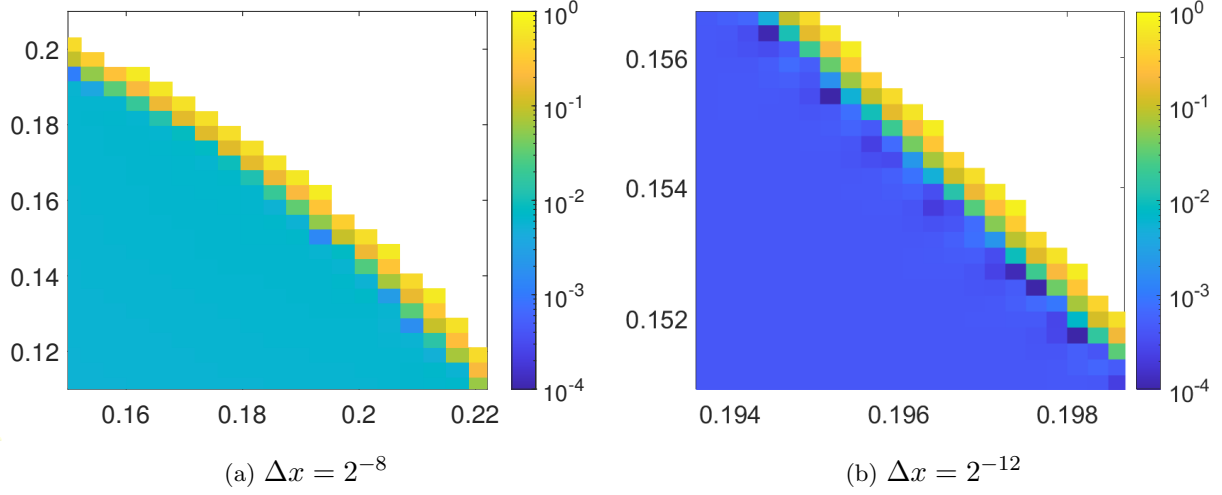
FIGURE 5.2. Plots of errors from using IBDL method to solve Equation (5.1) with $k = 1$ and $g = 0$ on the periodic computational domain $[-0.5, 0.5]^2$, where the circular boundary of radius 0.25 has prescribed boundary values given by $U_b = e^x$. The errors are shown for two grid point spacings and with $\Delta s \approx \Delta x$. The figure is zoomed-in to view the large errors that lie in a small region near Γ.

The full IBDL method for Poisson and Helmholtz equations can now be described in the following steps:

(1) Use a Krylov method to solve the following system for Q.

$$-(S^* \mathcal{L}^{-1} \widetilde{S})Q + \frac{1}{2}Q = U_b - S^* \mathcal{L}^{-1} \tilde{g} \tag{5.14}$$

(2) Use Q to solve the following equation for u in $\mathcal{C}$.

$$\mathcal{L}u + \widetilde{S}Q = \tilde{g} \tag{5.15}$$

(3) Use an interpolation to replace the values of u at $\boldsymbol{x} \in \Omega$ that are near Γ. Factors that determine the number of meshwidths to include in this step are discussed in Section 5.5.

5.1.3   Arbitrary immersed boundary parametrization

In this dissertation, we utilize an arclength parametrization of the immersed boundary Γ. Numerically, this means that in the discretization of the spread operator, Δs_i is the size of the arclength connecting $\boldsymbol{X}(s_i)$ and $\boldsymbol{X}(s_{i+1})$. In practice, this choice is not a limiting factor because even if an

59

analytical arclength parametrization of the immersed boundary curve is not available, one may still approximate this value with the distance $\Delta s_i \approx ||\boldsymbol{X}(s_{i+1}) - \boldsymbol{X}(s_i)||_2$. However, to be thorough, in this section, we discuss the use of an arbitrary parametrization.

Let $\boldsymbol{X}(s)$ be an arbitrary parametrization of Γ. Using this parametrization and the derivations from Chapter 3, the double layer integral equation for Poisson would have the form

$$U_b(\boldsymbol{X}(s')) = \int_\Gamma^{PV} \gamma(\boldsymbol{X}(s))\nabla G(\boldsymbol{X}(s), \boldsymbol{X}(s')) \cdot \boldsymbol{n}(\boldsymbol{X}(s))\left|\frac{\partial \boldsymbol{X}}{ds}\right|ds - \frac{1}{2}\gamma(\boldsymbol{X}(s')). \tag{5.16}$$

In this case, we can form the IBDL method as

$$\mathcal{L}u + \widetilde{S}Q = \tilde{g} \qquad \text{in } \mathcal{C} \tag{5.17a}$$

$$S^*u + \frac{1}{2|\partial \boldsymbol{X}/\partial s|}Q = U_b \qquad \text{on } \Gamma. \tag{5.17b}$$

Then, using the same steps as in Section 5.1.1, we would obtain the equation

$$U_b(\boldsymbol{X}(s')) = -\int_\Gamma^{PV} Q(\boldsymbol{X}(s))\nabla G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s')) \cdot \boldsymbol{n}(\boldsymbol{X}(s))ds + \frac{1}{2|\partial \boldsymbol{X}/\partial s|}Q(\boldsymbol{X}(s')). \tag{5.18}$$

By comparing Equations (5.16) and (5.18), we see that in the limit $h \longrightarrow 0$, we get

$$Q(\boldsymbol{X}) = -\gamma(\boldsymbol{X})\left|\frac{\partial \boldsymbol{X}}{\partial s}\right|. \tag{5.19}$$

As discussed in Section 3.7, the constant $1/2$ shifts all of the eigenvalues of the integral operator so that the only limit point is away from 0. This ensures a small, constant condition number as the mesh is refined and therefore small iteration counts for the Krylov method. An arbitrary paramatrization may not shift the eigenvalues in this uniform manner. As such, the iteration counts may increase. In Section 5.4.1, we will revisit the Helmholtz equation from Section 2.4.1 given by

$$\Delta u - u = 0 \qquad \text{in } \Omega \tag{5.20a}$$

$$u = \sin 2\theta \qquad \text{on } \Gamma, \tag{5.20b}$$

where Ω is the interior of a circle of radius 0.25, centered at the origin. We will see that the iteration count for solving this equation with the IBDL method using arclength parametrization is about 4-5 iterations. If we instead use the parametrization given by

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.25 \cos\left(s^2 + s\right) \\ 0.25 \sin\left(s^2 + s\right) \end{pmatrix}, \tag{5.21}$$

for $0 \leq s \leq \frac{1}{2}(-1 + \sqrt{1 + 8\pi})$, we have

$$\left| \frac{\partial \boldsymbol{X}}{\partial s} \right| = 0.25(2s + 1). \tag{5.22}$$

Using immersed boundary points that are equally spaced in the parameter s, the iteration count is about 20. The iteration count is larger, but it does remain constant as the mesh is refined. This is likely to be the case with an arbitrary parametrization as the eigenvalues will still be shifted in some manner away from 0. It is therefore acceptable to implement the IBDL method in this manner, but we then need the values or approximations of $|\partial \boldsymbol{X}/\partial s|$. To maintain consistency in the formulation of the IBDL method, we will use an arclength parametrization, with equally spaced points, in all other sections of this dissertation.

5.2   Neumann boundary conditions

We now use the IB framework and our connection to boundary integral equations to solve a PDE with Neumann boundary conditions. The original IBSL method is unable to handle such a problem since the solution derivatives are not convergent at the boundary.

Let us look at the PDE

$$\mathcal{L}u = g \qquad \text{in } \Omega \tag{5.23a}$$

$$\frac{\partial u}{\partial n} = V_b \qquad \text{on } \Gamma, \tag{5.23b}$$

where we will first take $g = 0$ for simplicity. In Section 3.3, we introduced the full integral representation of u. For $\boldsymbol{x}_0 \in \Omega \setminus \Gamma$, we have

$$u(\boldsymbol{x}_0) = \int_\Gamma \nabla u(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) G(\boldsymbol{x}, \boldsymbol{x}_0) dl(\boldsymbol{x}) - \int_\Gamma u(\boldsymbol{x}) \nabla G(\boldsymbol{x}, \boldsymbol{x}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}). \tag{5.24}$$

Then, we can take $\boldsymbol{x}_0 \longrightarrow \Gamma$ in Equation (5.24) and denote it as $\boldsymbol{X}_0$. Using the property of the Green's function derivative from Section 3.1, we get

$$u(\boldsymbol{X}_0) = \int_\Gamma \nabla u(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) G(\boldsymbol{x}, \boldsymbol{X}_0) dl(\boldsymbol{x}) - \int_\Gamma^{PV} u(\boldsymbol{x}) \nabla G(\boldsymbol{x}, \boldsymbol{X}_0) \cdot \boldsymbol{n}(\boldsymbol{x}) dl(\boldsymbol{x}) + \frac{1}{2} u(\boldsymbol{X}_0). \tag{5.25}$$

The first term in Equation (5.25) is a single layer potential with the strength given by the known boundary derivatives, V_b. The second term is a double layer potential whose strength is given by the *unknown* boundary values. If we let $U_b \equiv u|_\Gamma$ be the unknown boundary values, we can then write the PDE in the IB framework as

$$\mathcal{L}u + \widetilde{S}U_b + SV_b = 0 \tag{5.26}$$

Then, recall that the second IBDL equation in the Dirichlet case is given by

$$S^* u + \frac{1}{2}Q = U_b. \tag{5.27}$$

In the Neumann case, since U_b is our unknown boundary distribution, we replace Q in Equation (5.27) with U_b, and we get the corresponding equation in the Neumann case. Allowing for a non-zero g, we can therefore write the PDE in the Immersed Boundary framework as

$$\mathcal{L}u + \widetilde{S}U_b + SV_b = \tilde{g} \qquad \text{in } \mathcal{C} \tag{5.28a}$$

$$S^* u = \frac{1}{2}U_b \qquad \text{on } \Gamma. \tag{5.28b}$$

For Neumann boundary conditions, U_b is the *unknown* potential strength on the boundary, corresponding to the unknown boundary values.

We can solve this system by first solving the equation

$$(-S^*\mathcal{L}^{-1}\widetilde{S})U_b - \frac{1}{2}U_b = S^*\mathcal{L}^{-1}SV_b - S^*\mathcal{L}^{-1}\tilde{g} \tag{5.29}$$

for the boundary values, U_b, and then obtaining u from Equation (5.28a). We can see by comparing to Equation (5.14) that the operator is similar to that of the IBDL method for Dirichlet boundary conditions, where the only difference is the sign on the $1/2$. Therefore, we again get a well-conditioned problem that can be solved with a small number of iterations of a Krylov method.

We now provide a more detailed demonstration of the connection between Equation (5.25) and Equation (5.28). We use steps similar to those used in Sections 4.1.2 and 5.1.1.

Starting with the Equation (5.28a) and using $\tilde{g} = 0$ for simplicity, we have

$$\mathcal{L}u = -SV_b - \widetilde{S}U_b = -\int_\Gamma V_b(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds - \nabla \cdot \int_\Gamma U_b(s)\boldsymbol{n}(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{5.30}$$

Inverting the operator and using that $\mathcal{L}G_h(\boldsymbol{x}, \boldsymbol{x}_0) = -\delta_h(\boldsymbol{x} - \boldsymbol{x}_0)$, we get

$$u(\boldsymbol{x}) = \int_\Gamma V_b(s)G_h(\boldsymbol{x}, \boldsymbol{X}(s))ds + \nabla \cdot \int_\Gamma U_b(s)\boldsymbol{n}(s)G_h(\boldsymbol{x}, \boldsymbol{X}(s))ds \tag{5.31}$$

for $\boldsymbol{x} \in \Omega \setminus \Gamma$. Bringing in the divergence and manipulating the expression, we get

$$u(\boldsymbol{x}) = \int_\Gamma V_b(s)G_h(\boldsymbol{x}, \boldsymbol{X}(s))ds + \int_\Gamma U_b(s)\nabla G_h(\boldsymbol{x}, \boldsymbol{X}(s)) \cdot \boldsymbol{n}(s)ds. \tag{5.32}$$

The second equation of the method, Equation (5.28b), gives us

$$\frac{1}{2}U_b(s') = \int_\mathcal{C} u(\boldsymbol{x})\delta_h(\boldsymbol{x} - \boldsymbol{X}(s'))dx. \tag{5.33}$$

Combining Equation (5.32) with Equation (5.33), changing the order of integration, and recognizing the presence of $G_{hh} = G_h * \delta_h$, we get

$$\frac{1}{2}U_b(s') = \int_\Gamma V_b(s)G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s))ds + \int_\Gamma U_b(s)\nabla G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) \cdot \boldsymbol{n}(s)ds. \tag{5.34}$$

Note that due to the presence of a double layer potential, as discussed in Section 5.1.1, we assume that $\boldsymbol{X}(s)$ is the arclength parametrization of Γ so that $\left|\partial \boldsymbol{X}(s)/\partial s\right| = 1$.

The symmetry of the Green's function, which is preserved through convolutions with the regularized delta function, gives us that $G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) = G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))$. Additionally, the odd symmetry of the gradient of the Green's function gives us $\nabla G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) = -\nabla G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))$ [51]. Using these to switch the arguments of these functions, we get

$$\frac{1}{2}U_b(s') = \int_\Gamma V_b(s)G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))ds - \int_\Gamma U_b(s)\nabla G_{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s')) \cdot \boldsymbol{n}(s)ds. \qquad (5.35)$$

Appropriately redefining U_b, V_b, and $\boldsymbol{n}$ as functions of $\boldsymbol{x}$, we get

$$\frac{1}{2}U_b(\boldsymbol{X}(s')) = \int_\Gamma V_b(\boldsymbol{X}(s))G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s))ds$$
$$- \int_\Gamma U_b(\boldsymbol{X}(s))\nabla G_{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) \cdot \boldsymbol{n}(\boldsymbol{X}(s))ds. \quad (5.36)$$

Using our arclength parametrization in Equation (5.25) and combining the $u(\boldsymbol{X}_0)$ terms, we can write the integral equation for $\boldsymbol{X}_0$ on the boundary as

$$\frac{1}{2}u(\boldsymbol{X}(s')) = \int_\Gamma \nabla u(\boldsymbol{X}(s)) \cdot \boldsymbol{n}(\boldsymbol{X}(s))G(\boldsymbol{X}(s), \boldsymbol{X}(s'))ds$$
$$- \int_\Gamma^{PV} u(\boldsymbol{X}(s))\nabla G(\boldsymbol{X}(s), \boldsymbol{X}(s')) \cdot \boldsymbol{n}(\boldsymbol{X}(s))ds. \quad (5.37)$$

By comparing Equations (5.36) and (5.37), we can see that the IB formulation presented in Equation (5.28) is equivalent to a regularized integral equation.

5.3 Numerical implementation

We now discuss the numerical implementation of the IBDL method, particularly the areas in which the method differs from the IBSL method. There is no change in the discretization of space, the differential operators, and the basic spread and interpolation operators, and this information can be found in Section 2.2.

5.3.1 Discretization of spread operator

We will first discuss the implementation of the operator $\widetilde{S}$ defined by

$$\widetilde{S}Q = \nabla \cdot (SQ\boldsymbol{n}). \tag{5.38}$$

This operation consists of first multiplying our discretized boundary density Q by the outward unit normal vectors at the immersed boundary points. We then spread these vectors to the grid and take the divergence. We discuss the implementation of each of these steps below.

The first component of the operator $\widetilde{S}$ is the divergence. To compute this, we use either a finite difference or Fourier spectral method to match the method used to discretize the PDE. For finite differences, we use the standard second-order accurate centered difference for derivatives, and for the Fourier spectral method, we use the derivatives discussed in Section 2.3.1.

Next, we need the unit normal vectors to the immersed boundary. These can be treated as an input to the method, or we can approximate them if they are unavailable. Approximating the curve with linear elements and using the resulting normal vectors is sufficient for maintaining first-order accuracy. For instance, for an interior domain and a counterclockwise curve parametrization, we can use

$$\boldsymbol{n}(s_i) = \begin{bmatrix} Y(s_{i+1}) - Y(s_{i-1}) \\ -\big(X(s_{i+1}) - X(s_{i-1})\big) \end{bmatrix} \Big/ \big|\big|\boldsymbol{X}(s_{i+1}) - \boldsymbol{X}(s_{i-1})\big|\big|_2. \tag{5.39}$$

For an exterior domain, we would negate this.

We next look at the spread operator S, which is defined as

$$(SF)(\boldsymbol{x}) = \int_\Gamma F(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{5.40}$$

As discussed in Section 2.2, we use the traditional Peskin four-point delta function for δ_h [**47**], unless otherwise specified.

Then, as mentioned in Section 5.1, we discretize the integral in Equation (5.40) with respect to arclength, so henceforth, I will treat s as an arclength parameter. With this new assumption on the coordinate s, Δs_i gives the length of the curve connecting the immersed boundary points,

65

$\boldsymbol{X}(s_i)$ and $\boldsymbol{X}(s_{i+1})$. Then, the discrete spread operator is given by

$$(SF)(\boldsymbol{x}) = \sum_{i=1}^{N_{IB}} F(s_i)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s_i))\Delta s_i. \tag{5.41}$$

Depending on the parametrization, the boundary points may not be equally spaced, but since we will be looking at a periodic distribution on Γ, choosing equally spaced points will give spectral accuracy in our integral approximation. Unless otherwise specified, we will space the boundary points equally with $\Delta s \approx \alpha \Delta x$ for various values of α. Then $\Delta s = L_{IB}/N_{IB}$, where L_{IB} gives the length of the immersed boundary. If the exact values of Δs_i are unavailable, approximations can be made. Again, using $\Delta s_i \approx ||\boldsymbol{X}(s_{i+1}) - \boldsymbol{X}(s_i)||_2$ is sufficient to maintain first-order accuracy.

5.3.2   Solution to discrete system for invertible $\mathcal{L}$

We now look at the discretized system

$$\mathcal{L}u + \widetilde{S}Q = \tilde{g} \qquad \text{in } \mathcal{C} \tag{5.42a}$$

$$S^*u + \frac{1}{2}Q = U_b \qquad \text{on } \Gamma. \tag{5.42b}$$

In the case that the differential operator is invertible, such as for $\mathcal{L} = \Delta - k^2$, for $k \neq 0$, we can invert the operator to obtain

$$u = -\mathcal{L}^{-1}\widetilde{S}Q + \mathcal{L}^{-1}\tilde{g}. \tag{5.43}$$

Then, by applying the interpolation operator and using Equation (5.42b), we obtain

$$-(S^*\mathcal{L}^{-1}\widetilde{S})Q + \frac{1}{2}Q = U_b - S^*\mathcal{L}^{-1}\tilde{g}. \tag{5.44}$$

We can therefore solve Equation (5.42) by first solving Equation (5.44) for Q and then obtaining u from Equation (5.42a). Here, the operator that must be inverted, $-(S^*\mathcal{L}^{-1}\widetilde{S}) + \frac{1}{2}\mathbb{I}$, is not symmetric, and we therefore solve for Q with `gmres`, to a tolerance of 10^{-8}, unless otherwise specified.

5.3.3 Solution to discrete system for Poisson equation

In the case that the differential operator is the periodic Laplacian, $\mathcal{L} = \Delta$, the solution method outlined in the previous section must be adjusted, both for the nullspace of the discrete constraint system and the nullspace of $\mathcal{L}$.

Nullspace of Constraint System. In the case of the IBSL method applied to the Poisson equation, the constraint matrix, given by

$$\begin{pmatrix} \Delta & S \\ S^* & 0 \end{pmatrix}, \tag{5.45}$$

is invertible. By handling the nullspace of Δ as described in Section 2.2.4, we can therefore recover the unique solution to the constraint system. In the case of the IBDL method, however, the constraint matrix, given by

$$\begin{pmatrix} \Delta & \widetilde{S} \\ S^* & \frac{1}{2}\mathbb{I} \end{pmatrix}, \tag{5.46}$$

has a one-dimensional nullspace. The analytical nullspace is spanned by

$$u = \chi_{\mathcal{C}\backslash\Omega} \tag{5.47a}$$

$$Q = -1, \tag{5.47b}$$

where $\chi_{\mathcal{C}\backslash\Omega}$ gives the indicator function that is 0 on Ω and 1 otherwise. Section 5.3.5 demonstrates the use of a similar solution to create an indicator function. According to Equation (5.47), since an element of the nullspace would be 0 on the physical domain, any solution we get would be the same on Ω, and this nullspace would therefore not pose a problem for the IBDL method. We have observed this to be true for interior domains or small exterior domains. However, we have observed that for some boundary value problems on exterior domains for which the proportion of the periodic box occupied by Ω is large, the IBDL method outlined does not converge to the correct solution. We conjecture that this may be explained by two facts. Firstly, while Equation (5.47) gives the *analytical* nullspace, our regularization and discretization results in a different numerical nullspace.

Additionally, in the infinite exterior domain case, the decay of the double layer potential means that it is unable to represent solutions to the Poisson equation with arbitrary boundary values [**53**]. As we increase the periodic domain length L, we may therefore approach an analytical problem that does not have a solution, and the contribution from an element of our numerical nullspace may be larger. We give a solution to this issue below, but more investigation is needed into the subject of this nullspace.

For large exterior domains, therefore, we may use the completed double layer representation, which was discussed in Section 3.5.3, as it relates to Stokes equation. We can formulate this completed representation by the addition of a single layer potential whose strength is a constant multiple of the strength of the double layer potential. The completed system is then given by

$$\Delta u + \eta S Q + \widetilde{S} Q = \tilde{g} \qquad \text{in } \mathcal{C} \tag{5.48a}$$

$$S^* u + \frac{1}{2} Q = U_b \qquad \text{on } \Gamma, \tag{5.48b}$$

where η is a positive constant. Appropriate choices for η are discussed in Section 6.4.2. We will refer to this form of the IBDL method as the *completed IBDL method*. In this case, the constraint matrix, given by

$$\begin{pmatrix} \Delta & \widetilde{S} + \eta S \\ S^* & \frac{1}{2}\mathbb{I} \end{pmatrix}, \tag{5.49}$$

is invertible, and then we need only deal with the nullspace of the periodic Laplacian. In Section 5.4.3, we examine a problem for which the completed IBDL method is needed.

Nullspace of $\mathcal{L}$. As discussed in Section 2.2.4, the original boundary value problem on Ω has a unique solution, but, by using the IB framework, we embed the PDE into a periodic computational domain on which Δ is not invertible.

As we did in Section 2.2.4, let us begin by decomposing the solution u as

$$u = u_0 + \bar{u}, \tag{5.50}$$

where u_0 has mean 0 on $\mathcal{C}$ and $\bar{u}$ gives the mean value of u on $\mathcal{C}$. The PDE on $\mathcal{C}$ that results from the completed IBDL formulation in Equation (5.48) is given by

$$\Delta u + \eta SQ + \nabla \cdot (SQ\boldsymbol{n}) = \tilde{g}, \tag{5.51}$$

where $\tilde{g} \equiv g\chi_\Omega + g_e\chi_{\mathcal{C}\backslash\Omega}$ is an extension of g. To derive the solvability condition, we integrate Equation (5.51) over a general computational domain $\mathcal{C}$. Noting again that our spread operator is defined using a regularized delta function, the functions are smooth, and we can use the divergence theorem. Let us again use $\partial\mathcal{C}$ as the boundary of $\mathcal{C}$ and $\boldsymbol{n}_\mathcal{C}$ as the unit normal on $\partial\mathcal{C}$, to distinguish it from $\boldsymbol{n}$, which we continue to use as the unit normal on the immersed boundary Γ. We then get

$$\int_{\partial\mathcal{C}} \nabla u \cdot \boldsymbol{n}_\mathcal{C} dl(\boldsymbol{x}) + \eta \int_\mathcal{C} SF d\boldsymbol{x} + \int_{\partial\mathcal{C}} SQ\boldsymbol{n} \cdot \boldsymbol{n}_\mathcal{C} dl(\boldsymbol{x}) = \int_\mathcal{C} \tilde{g} d\boldsymbol{x}. \tag{5.52}$$

If Γ is away from $\partial\mathcal{C}$, the third term vanishes due to the compact support of the integrand. Then, if we take $\mathcal{C}$ to be the periodic box used in this work, the first term also disappears. Using that $\int_\mathcal{C} SF d\boldsymbol{x} = \int_\Gamma F ds$, the solvability condition on the periodic computational domain $\mathcal{C}$ then has the form

$$\eta \int_\Gamma Q ds = \int_\mathcal{C} \tilde{g}(\boldsymbol{x}) d\boldsymbol{x}. \tag{5.53}$$

When we are not using the completed formulation, $\eta = 0$, and we can easily satisfy this condition with our choice of g_e. For example, we can choose

$$g_e(\boldsymbol{x}) = -\frac{1}{|\mathcal{C} \backslash \Omega|} \int_\Omega g(\boldsymbol{x}) d\boldsymbol{x}, \tag{5.54}$$

where we use $|\cdot|$ to denote the area of the domain. With this constraint satisfied, the solution to Equation (5.51) can be found, and it is unique up to an additive constant. Therefore, since we are not interested in the solution on the nonphysical domain, $\mathcal{C} \backslash \Omega$, we can use the solution with mean 0 on $\mathcal{C}$. If we select a solution with a different mean, the potential strength Q will adjust to enforce the boundary condition on the physical side of Γ, maintaining the unique solution on the PDE domain Ω.

If, on the other hand, we use the completed IBDL method, we can use the constraint in Equation (5.53) to find the value of $\bar{u}$, as we did in the IBSL method in Section 2.2.4. Discretizing

69

the constraint, we get the following system of equations.

$$\Delta u_0 + \eta SQ + \widetilde{S}Q = \tilde{g} \tag{5.55a}$$

$$S^* u_0 + \bar{u} \mathbb{1}_{N_{IB}} + \frac{1}{2}Q = U_b \tag{5.55b}$$

$$\eta(\Delta s)\mathbb{1}_{N_{IB}}^\mathsf{T} Q = (\Delta x \Delta y)\mathbb{1}_{N^2}^\mathsf{T} \tilde{g}. \tag{5.55c}$$

Using the operator previously defined to return a solution with mean 0 on $\mathcal{C}$, we use `gmres` to solve the following system of discrete equations for Q and $\bar{u}$.

$$- (S^* \Delta_0^{-1} \widetilde{S})Q - \eta(S^* \Delta_0^{-1} S)Q + \bar{u}\mathbb{1}_{N_{IB}} + \frac{1}{2}Q = U_b - S^* \Delta_0^{-1} \tilde{g} \tag{5.56a}$$

$$\eta(\Delta s)\mathbb{1}_{N_{IB}}^\mathsf{T} Q = (\Delta x \Delta y)\mathbb{1}_{N^2}^\mathsf{T} \tilde{g} \tag{5.56b}$$

We can then obtain u by computing

$$u = -\Delta_0^{-1}(\widetilde{S} + \eta S)Q + \Delta_0^{-1}\tilde{g} + \bar{u}\mathbb{1}_{N^2}. \tag{5.57}$$

5.3.4   Interpolation for near-boundary points

As discussed in Section 5.1.2, we use a linear interpolation to replace solution values within m_1 meshwidths of the boundary Γ. For the Dirichlet problem, the interpolation for an individual gridpoint uses one approximate solution value located about m_2 meshwidths away from Γ and one value approximated from the known solution values located on Γ. For the Neumann problem, the solution values located on Γ are not known, and therefore they are also approximate. In this section, we discuss the details of this linear interpolation.

Figure 5.3 illustrates the interpolation for one point, $\boldsymbol{x}_p$. Given such a point, we first locate the two nearest boundary points, labeled in the picture as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. We use a projection to find $\boldsymbol{x}_A$, the point on the segment between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ that is closest to $\boldsymbol{x}_p$. It is given by

$$\boldsymbol{x}_A = \boldsymbol{x}_2 + \frac{(\boldsymbol{x}_2 - \boldsymbol{x}_p) \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2)}{||\boldsymbol{x}_1 - \boldsymbol{x}_2||^2}(\boldsymbol{x}_1 - \boldsymbol{x}_2). \tag{5.58}$$
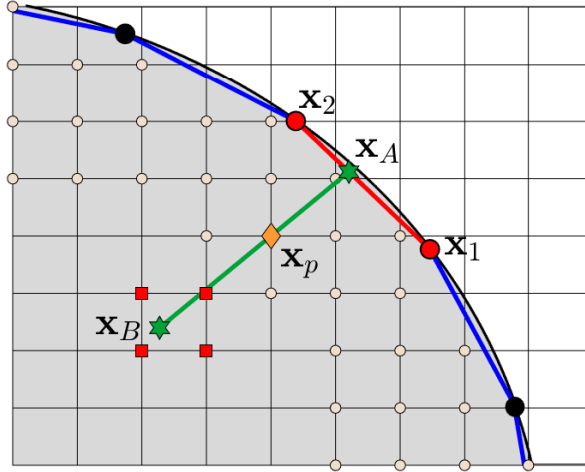
FIGURE 5.3. Illustration of the interpolation step of the IBDL method. The PDE domain Ω is shown in grey. The exact boundary curve is shown in black with black circles indicating immersed boundary points. Lines connecting the IB points are shown in blue. Grid points within $m_1 = 3$ meshwidths of the boundary are indicated by beige circles. The diagram illustrates the process for interpolating the value for one point, denoted with an orange diamond and labeled $\boldsymbol{x}_p$. The two closest boundary points are marked with red circles and labeled $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The closest point on the line segment connecting $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is called $\boldsymbol{x}_A$ and is marked with a green pentagram. Also marked with a green pentagram is the point m_2 meshwidths away from $\boldsymbol{x}_A$, and it is labeled $\boldsymbol{x}_B$. Marked with red squares are the four closest grid points to $\boldsymbol{x}_B$.

If the curvature and point placements are such that this calculation results in a point not located on the line segment between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we simply find the third closest boundary point and use the two outermost points to find $\boldsymbol{x}_A$. Next, we use a simple interpolation of the values at $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ to approximate the value at $\boldsymbol{x}_A$, and this is given by

$$u(\boldsymbol{x}_A) \approx U_b(\boldsymbol{x}_1)\frac{||\boldsymbol{x}_A - \boldsymbol{x}_2||}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||} + U_b(\boldsymbol{x}_2)\frac{||\boldsymbol{x}_A - \boldsymbol{x}_1||}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||}, \tag{5.59}$$

where the values at $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are denoted with U_b because these are known boundary values.

We next find the second point of interpolation, located m_2 meshwidths away from point $\boldsymbol{x}_A$. It is labeled in the figure as $\boldsymbol{x}_B$, and it is given by

$$\boldsymbol{x}_B = \boldsymbol{x}_A + \frac{\boldsymbol{x}_p - \boldsymbol{x}_A}{||\boldsymbol{x}_p - \boldsymbol{x}_A||}(m_2\Delta x). \tag{5.60}$$

Note that this step assumes that points are labeled as interior or exterior according to the *discretized* shape created by the IB points. One way of doing this is presented in Section 5.3.5. Otherwise,

if we are completing this interpolation for a point $\boldsymbol{x}_p$ that is in the interior of the analytical boundary shape but exterior to the discretized shape, we would need to alter Equation (5.60) to use subtraction instead of addition.

We then estimate the value for $\boldsymbol{x}_B$ with a bilinear interpolation of the computed solution values at the four grid points located around $\boldsymbol{x}_B$, marked in the figure with red squares. Let $\boldsymbol{x}_{00}$ be the lower left point, $\boldsymbol{x}_{01}$ be the upper left, and so on, and let d_x and d_y be the x and y distances from $\boldsymbol{x}_B$ to $\boldsymbol{x}_{00}$. Then our bilinear interpolation is

$$
\begin{aligned}
u(\boldsymbol{x}_B) \approx & \left(\frac{d_x}{\Delta x}\right)\left(\frac{d_y}{\Delta y}\right) u(\boldsymbol{x}_{11}) + \left(\frac{d_y}{\Delta y}\right)\left(1 - \frac{d_x}{\Delta x}\right) u(\boldsymbol{x}_{01}) \\
& + \left(\frac{d_x}{\Delta x}\right)\left(1 - \frac{d_y}{\Delta y}\right) u(\boldsymbol{x}_{10}) + \left(1 - \frac{d_x}{\Delta x} - \frac{d_y}{\Delta y} + \left(\frac{d_x}{\Delta x}\right)\left(\frac{d_y}{\Delta y}\right)\right) u(\boldsymbol{x}_{00}).
\end{aligned} \quad (5.61)
$$

Note that if $\boldsymbol{x}_B$ is located directly on a grid line, this can be simplified.

Once we have the approximated values at $\boldsymbol{x}_B$ and $\boldsymbol{x}_A$, we again use a simple linear interpolation to approximate $u(\boldsymbol{x}_p)$, and this is given by

$$
u(\boldsymbol{x}_p) \approx u(\boldsymbol{x}_A) \frac{||\boldsymbol{x}_B - \boldsymbol{x}_p||}{||\boldsymbol{x}_B - \boldsymbol{x}_A||} + u(\boldsymbol{x}_B) \frac{||\boldsymbol{x}_A - \boldsymbol{x}_p||}{||\boldsymbol{x}_B - \boldsymbol{x}_A||}. \quad (5.62)
$$

5.3.5   Flagging interior and exterior points using the IBDL method

In order to complete the interpolation step and to isolate the solution on Ω, it is necessary to identify grid points as interior or exterior to Γ. Depending on the boundary shape, this can be done with the exact parametrization of the curve, but we can also use the IBDL method itself to identify grid points as interior to the *discretized* curve. In this section, we show that by using analytical features of the indicator function for the interior of the immersed boundary Γ, we arrive at a familiar IBDL expression that enables us to use the framework already set forth in order to flag grid points as either exterior or interior to Γ. Eldredge also discusses this in Appendix A.2 of [**17**], and his work uses such an indicator function as a foundation on which to build PDEs that govern different variables on different sides of the immersed boundary. Here, we present the derivation and

application of this mechanism in the context of the IBDL method and also discuss how to apply it to a periodic computational domain.

Let the indicator function be defined as

$$\chi(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \text{ in } \Omega \\ 0 & \boldsymbol{x} \text{ in } \mathcal{C} \setminus \Omega, \end{cases} \tag{5.63}$$

where we assume here that Ω is the interior of Γ. Let $\psi \in C_c^\infty$ be a smooth, compactly supported test function. Then we have

$$\int_{\mathcal{C}} \chi \Delta \psi d\boldsymbol{x} = \int_{\Omega} \Delta \psi d\boldsymbol{x} = \int_{\Gamma} \nabla \psi \cdot \boldsymbol{n} dl(\boldsymbol{x}), \tag{5.64}$$

where we first use the definition of χ and then the divergence theorem. Alternatively, we can use integration by parts and the compact support of χ to obtain

$$\int_{\mathcal{C}} \chi \Delta \psi d\boldsymbol{x} = -\int_{\mathcal{C}} \nabla \chi \cdot \nabla \psi d\boldsymbol{x}, \tag{5.65}$$

where we can view $\nabla \chi$ as the weak derivative of χ. By equating the far right-hand side of Equation (5.64) and the right-hand side of Equation (5.65), we see that the weak derivative is the distribution given by

$$\nabla \chi = -\delta(\boldsymbol{x} - \boldsymbol{X})\boldsymbol{n}(\boldsymbol{x}), \tag{5.66}$$

where the $\boldsymbol{X}$ is a point on Γ, indicating that the the support of $\nabla \chi$ is Γ. This then gives us

$$-\Delta \chi = -\nabla \cdot \nabla \chi = \nabla \cdot (\delta \boldsymbol{n}), \tag{5.67}$$

and we therefore the PDE for χ given by

$$\Delta \chi + \nabla \cdot (\delta \boldsymbol{n}) = 0. \tag{5.68}$$

We can recognize $\nabla \cdot (\delta \boldsymbol{n})$ as the unregularized version of the term found in the IBDL method, $\widetilde{S}Q = \nabla \cdot (SQ\boldsymbol{n})$, where Q is a constant distribution of 1. This then gives us a way to construct

73

the indicator function χ by solving the PDE on $\mathcal{C}$ given by

$$\Delta\chi + \widetilde{S}\boldsymbol{n} = 0, \tag{5.69}$$

using the IBDL method.

Then, if we are using a periodic computational domain, as we do in this dissertation, the solution is only unique up to an additive constant. Therefore, we can first find the solution with mean 0 on $\mathcal{C}$. Then, to shift the exterior values to 0, we simply need to subtract the appropriate constant, which can be approximated using a value of χ far from Γ. A likely choice is the corner of the periodic domain. Subtracting this value from χ then results in a regularized indicator function that is approximately equal to $1/2$ on the boundary Γ. To retrieve the final indicator function, we simply set all values of χ that are less than $1/2$ equal to 0 and all those above $1/2$ to 1. Then χ is the indicator function described in Equation (5.63) that is equal to 1 for all points on the interior of the discretized boundary, Γ. Figure 5.4 gives an illustration of an indicator function resulting from applying this method to a set of three shapes. This process can be applied to multiple non-intersecting shapes at the same time. In other words, the PDE in Equation (5.69) needs only to be solved once, with the discrete spread operator applying to a distribution on the boundary points for all of the shapes.

5.4 Results: Dirichlet Helmholtz and Poisson equations

In this section, we apply the Immersed Boundary Double Layer method and discuss numerical results. In Section 5.4.1, we revisit the Dirichlet Helmholtz problem from Section 2.4.1 and demonstrate the improved efficiency of the IBDL method. In Section 5.4.2, we apply the method to a nonhomogeneous Poisson equation on a complex exterior domain. Lastly, in Section 5.4.3, we explore the need for the completed IBDL method for certain exterior domains
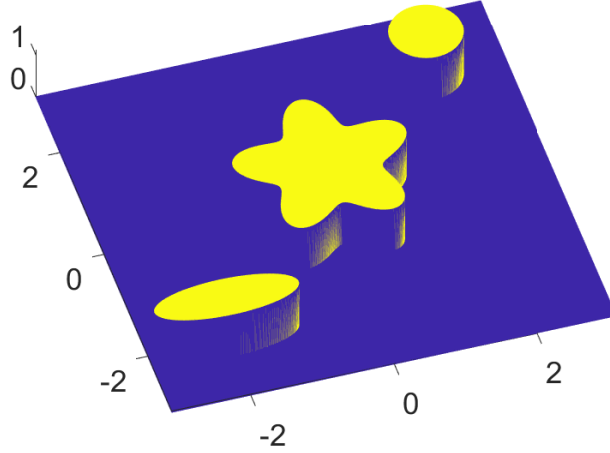
74

FIGURE 5.4. Illustration showing the indicator function resulting from the process described in Section 5.3.5 when applied to the discretized boundaries of three shapes.

5.4.1 Helmholtz equation revisited

First, we revisit the Helmholtz PDE from Section 2.4.1,

$$\Delta u - u = 0 \qquad \text{in } \Omega \tag{5.70a}$$

$$u = \sin 2\theta \qquad \text{on } \Gamma, \tag{5.70b}$$

where Ω is the interior of a circle of radius 0.25, centered at the origin. The analytical solution is given by

$$u = \frac{I_2(r) \sin 2\theta}{I_2(0.25)}, \tag{5.71}$$

where I_2 is the first-kind modified Bessel function of order 2. Our computational domain here is the periodic box $[-0.5, 0.5]^2$. For the IBDL method, we replace solution values within $m_1 = 6$ meshwidths of the boundary and use interpolation points located $m_2 = 8$ meshwidths from the boundary. We use equally spaced boundary points with $\Delta s \approx \alpha \Delta x$ for various values of α. Both methods use a finite difference method, and the solutions are computed for grid sizes ranging from $N = 2^6$ to 2^{12}.

75

| Iteration Counts - Circular Boundary							
	$\Delta s \approx 2\Delta x$		$\Delta s \approx 1.5\Delta x$		$\Delta s \approx 1\Delta x$		$\Delta s \approx 0.75\Delta x$	
Δx	IBSL	IBDL	IBSL	IBDL	IBSL	IBDL	IBSL	IBDL
2^{-6}	17	4	45	5	249	5	491	4
2^{-7}	36	5	57	4	691	5	752	4
2^{-8}	49	5	71	4	1233	4	3057	4
2^{-9}	61	4	110	4	1922	4	5829	4
2^{-10}	68	4	144	4	1936	4	8084	4
2^{-11}	95	4	234	4	4364	4	9335	4
2^{-12}	142	4	303	4	4535	4	10589	4

TABLE 5.1. Number of iterations of `minres` and `gmres`, with tolerance 10^{-8}, needed to solve Equation (5.70) using the IBSL and IBDL methods, respectively. Both methods use finite differences.

Table 5.1 gives the iteration counts for the Krylov methods used to solve Equation (5.70) for the IBSL and IBDL methods. This table illustrates the drastic improvement in efficiency that is a key advantage of the IBDL method. We see that only a small number of iterations are needed to solve for the strength of the IBDL potential. Moreover, the number of iterations remains essentially constant as we refine the mesh or tighten the boundary points relative to the grid. The IBSL method, on the other hand, requires many more iterations, and this iteration count is greatly affected by the mesh size and the boundary point spacing. Tighter boundary points can result in lower quadrature error, while more widely spaced points allow for smoother force distributions and better conditioning in the case of the IBSL method [35]. Making the IBSL method practical therefore often entails special handling, such as using more widely spaced points for discretization of the boundary, while using a denser set of points for the direct Eulerian-Lagrangian interaction [27]. Moreover, to make the IBSL method efficient, one must use a preconditioner [12, 30, 35, 54]. On the other hand, the fact that the IBDL method presented here can be efficiently utilized for both coarsely or finely spaced boundary points means no such step is required for this method.

In addition, the iteration counts for the IBSL method are affected by the tolerance for the Krylov method. For example, for $\Delta s \approx 2\Delta x$ and $\Delta x = 2^{-12}$, decreasing the tolerance from 10^{-8} to 10^{-10} increases the iteration count by more than fifty percent, from 142 to 233. For $\Delta s \approx 0.75\Delta x$, the Krylov method cannot reach a tolerance of 10^{-10} for the IBSL method. For the IBDL method, on the other hand, the Krylov method reaches this smaller tolerance in only 5 iterations for either of these cases.
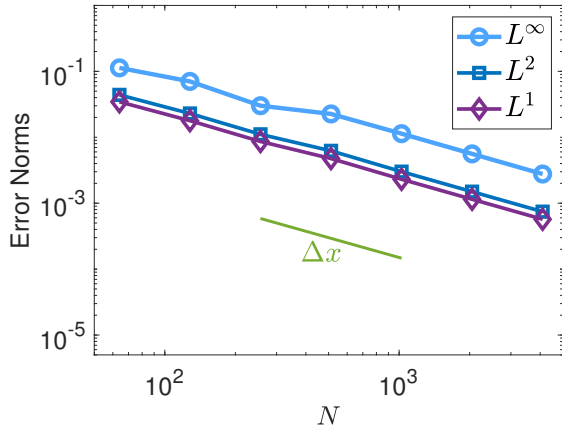
(a) IBSL Refinement
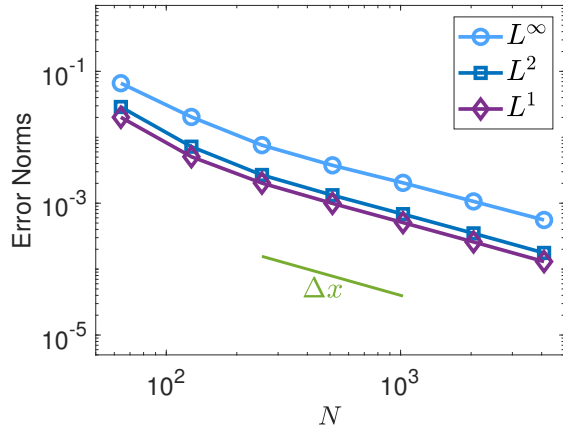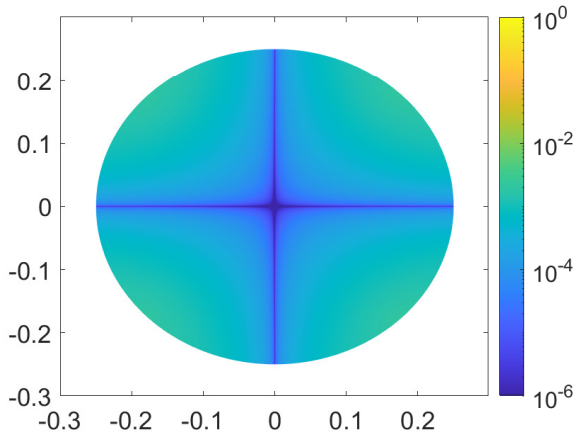
(b) IBDL Refinement
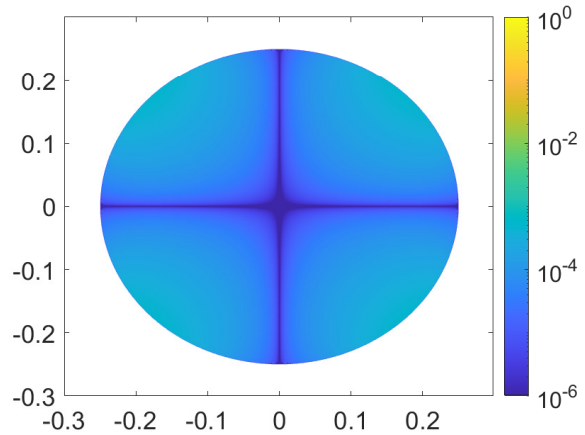
(c) IBSL Error

(d) IBDL Error

FIGURE 5.5. Error plots and refinement studies for solutions to Equation (5.70) found using the IBSL and IBDL methods, both with finite differences. The computational domain is the periodic box $[-0.5, 0.5]^2$, Ω is the interior of a circle of radius 0.25, and the prescribed boundary values are given by $U_b = \sin 2\theta$. Figures 5.5c-5.5d show the errors for the grid spacing $\Delta x = 2^{-12}$, and both methods use boundary point spacing of $\Delta s \approx 0.75\Delta x$. The IBDL method replaces solution values within $m_1 = 6$ meshwidths from the boundary using an interior interpolation point $m_2 = 8$ meshwidths away from the boundary. All plots use absolute errors.

Figures 5.5a-5.5b compare the IBSL and IBDL refinement studies, and we see that we maintain first-order convergence of the solution. Figures 5.5c-5.5d illustrate the solution errors for the two methods, and the errors from the IBDL method are actually lower for this problem. In general, the two methods typically give errors of comparable size.

Iteration Counts - Starfish boundary				
	$\Delta s \approx 2\Delta x$		$\Delta s \approx 0.75\Delta x$	
Δx	IBSL	IBDL	IBSL	IBDL
2^{-4}	57	13	1999	13
2^{-5}	84	13	5197	13
2^{-6}	87	13	4289	13
2^{-7}	86	13	3207	13
2^{-8}	74	14	2869	14
2^{-9}	65	14	2403	14
2^{-10}	55	14	1554	14

TABLE 5.2. Number of iterations of `minres` and `gmres`, with tolerance 10^{-8}, for the IBSL and IBDL methods, respectively, to solve Equation (5.72) on the periodic computational domain $[-2, 2]^2$. Ω is the region exterior to a "starfish" shape, and the prescribed boundary values are given by $U_b = \sin(\pi x/2) - \cos(\pi x/2)$. Both methods utilize finite differences.

5.4.2 Poisson equation on complex exterior domain

We move now to the Poisson equation for which we use the methods described in Sections 2.2.4 and 5.3.3 for the IBSL and IBDL methods, respectively, to handle the nullspace of the periodic Laplacian. Here we do not employ the completed double layer formulation, but use the original IBDL method as described in Equation (5.2). We also use this section to demonstrate the effectiveness of the IBDL method for a complex domain exterior to a non-convex boundary. We apply the IBSL and IBDL methods to the PDE given by

$$\Delta u = \frac{\pi^2}{4}\left(\cos\left(\frac{\pi y}{2}\right) - \sin\left(\frac{\pi x}{2}\right)\right) \qquad \text{in } \Omega \tag{5.72a}$$

$$u = \sin\left(\frac{\pi x}{2}\right) - \cos\left(\frac{\pi y}{2}\right) \qquad \text{on } \Gamma, \tag{5.72b}$$

where Ω is the region inside the periodic box $[-2, 2]^2$ that is exterior to the "starfish" shape used in [37], which is given by

$$\begin{pmatrix} x(\theta) \\ y(\theta) \end{pmatrix} = \left(1 + \frac{\sin(10\pi\theta)}{4}\right)\begin{pmatrix} \cos(2\pi\theta) \\ \sin(2\pi\theta) \end{pmatrix}, \tag{5.73}$$

for $0 \le \theta \le 1$. We use equally spaced boundary points, with $\Delta s \approx \alpha\Delta x$ for $\alpha = 0.75$ and 2. The solutions are computed for grid sizes ranging from $N = 2^6$ to 2^{14}, and the analytic solution is given by $u = \sin(\pi x/2) - \cos(\pi y/2)$. For the IBDL method, we replace solution values within
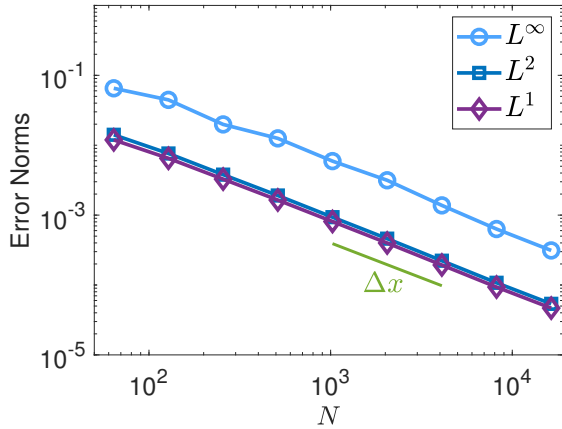
(a) IBSL Refinement
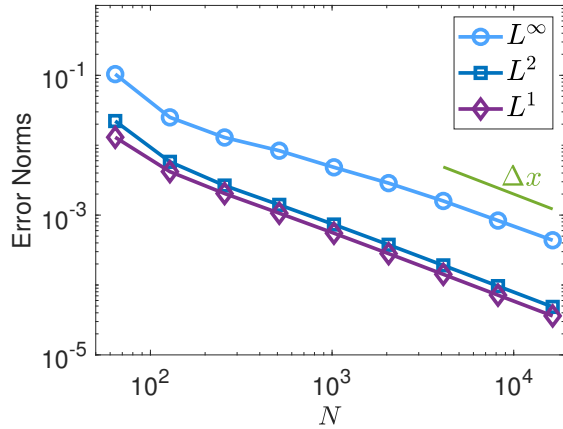
(b) IBDL Refinement

(c) IBSL Error

(d) IBDL Error

FIGURE 5.6. Refinement studies and error plots for solutions to Equation (5.72) found using the IBSL and IBDL methods, with finite dif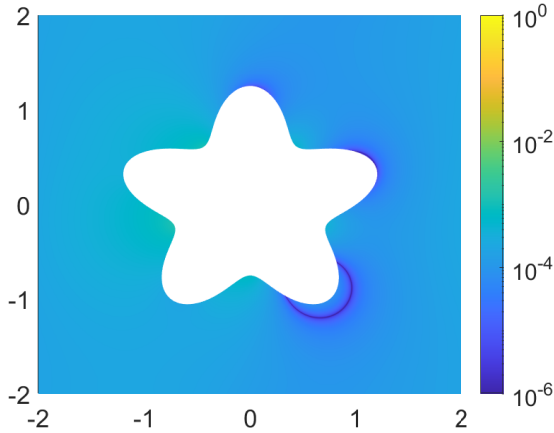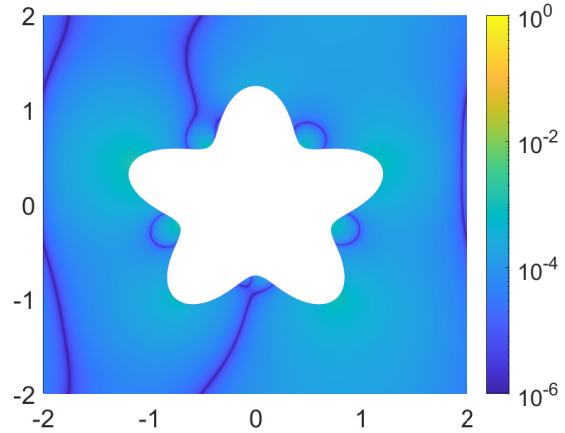ferences. The computational domain is the periodic box $[-2, 2]^2$, Ω is the region exterior to a "starfish" shape, and the prescribed boundary values are given by $U_b = \sin(\pi x/2) - \cos(\pi x/2)$. A boundary point spacing of $\Delta s \approx 0.75\Delta x$ is used, and in the IBDL method, solution values within $m_1 = 6$ meshwidths from the boundary are replaced using a second interpolation point $m_2 = 8$ meshwidths away from the boundary. Figures 5.6c-5.6d show plots of the solution errors for a mesh size of $N = 2^{12}$.

$m_1 = 6$ meshwidths of the boundary and use interpolation points located $m_2 = 8$ meshwidths from the boundary. Both methods utilize finite differences. Table 5.2 gives the iteration counts for the Krylov methods used by the IBSL and IBDL methods. Although the IBSL iteration counts do not always increase with grid refinement, the iteration counts are still higher than the IBDL iteration counts, particularly for $\Delta s \approx 0.75\Delta x$. Figure 5.6 shows the refinement studies and plots of the

solution errors for the IBSL and IBDL methods when $\alpha = 0.75$. Again, we see that the convergence is first-order for both methods, with comparable error sizes.

5.4.3   Completed IBDL method

In this section, we report some observations regarding the need for the completed IBDL method. We examine two PDEs to demonstrate that this need depends on both the size of the exterior domain and on the given boundary values. For both PDEs, Ω is the region in the periodic box $[-L/2, L/2]^2$ that is exterior to a circle of radius 0.25, centered at the origin. We will look at $L = 1$ and $L = 8$. The first PDE is

$$\Delta u = \frac{4\pi^2}{L^2} e^{\sin{(2\pi x/L)}} \Big(\cos^2{(2\pi x/L)} - \sin{(2\pi x/L)} \Big) \qquad \text{in } \Omega \tag{5.74a}$$

$$u = e^{\sin{(2\pi x/L)}} \qquad\qquad \text{on } \Gamma, \tag{5.74b}$$

and the analytical solution is given by $u = e^{\sin{(2\pi x/L)}}$. The second PDE is

$$\Delta u = -\frac{4\pi^2}{L^2} \Big(\sin{(2\pi x/L)} - \cos{(2\pi y/L)} \Big) \qquad \text{in } \Omega \tag{5.75a}$$

$$u = \sin{(2\pi x/L)} - \cos{(2\pi y/L)} \qquad \text{on } \Gamma, \tag{5.75b}$$

and the analytical solution is given by $u = \sin{(2\pi x/L)} - \cos{(2\pi y/L)}$. We use finite differences and interpolate values within $m_1 = 6$ meshwidths of the boundary. We first solve these problems using the IBDL method described in Equation (5.2), without the addition of the single layer potential. Figures 5.7a-5.7b illustrate that on the small periodic domain of length 1, both computed solutions have first-order convergence to the analytical solutions. However, Figures 5.7c-5.7d demonstrate that for a larger periodic domain of length 8, the two solutions behave very differently. The IBDL method is unable to achieve convergence to the analytical solution to Equation (5.74). Figure 5.7e shows that the error is essentially constant in the region away from the boundary. We may therefore be seeing the addition of an element of the nullspace of the periodic Laplacian to our solution. On the other hand, we see second-order convergence to the analytical solution for Equation (5.75).
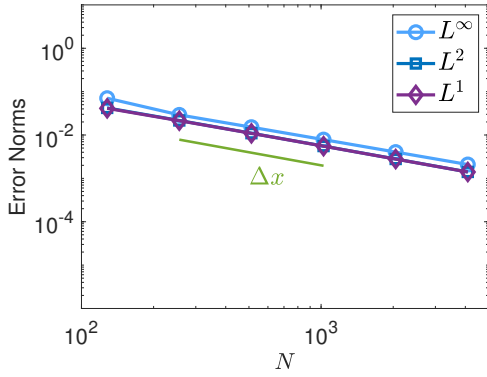
(a) Equation (5.74), $L = 1$
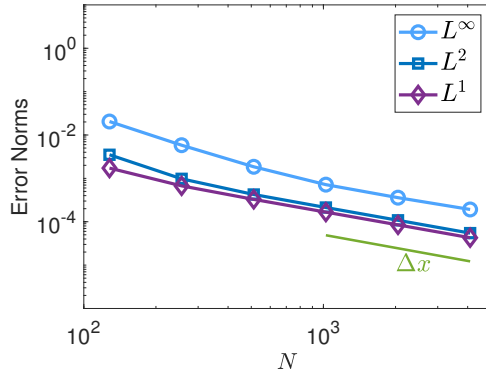
(b) Equation (5.75), $L = 1$
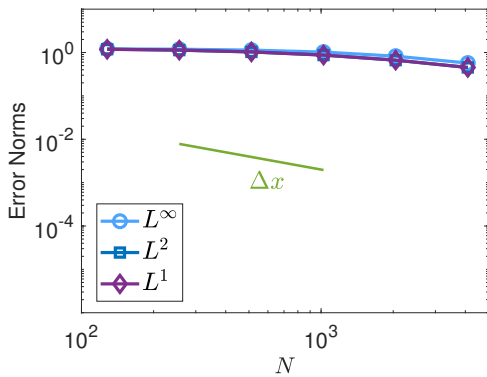
(c) Equation (5.74), $L = 8$
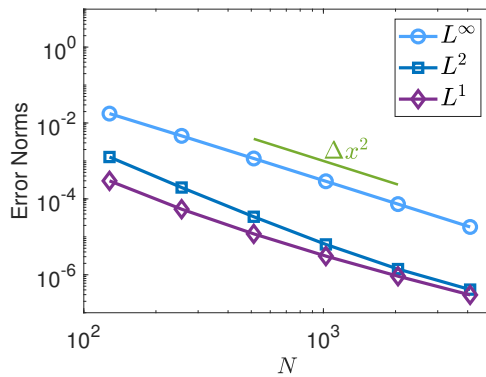
(d) Equation (5.75), $L = 8$
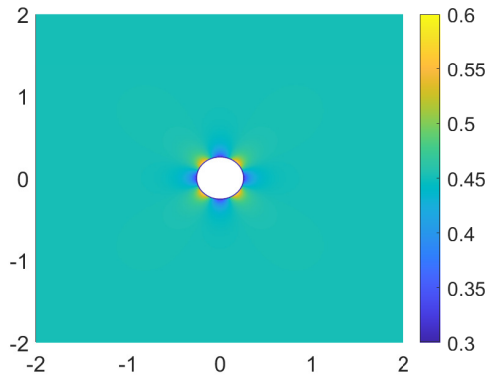
(e) Solution error, Equation (5.74), $L = 8$

FIGURE 5.7. Refinement studies for solutions to Equations (5.74)-(5.75) found using the IBDL method, with finite differences. For Figures 5.7a-5.7b, the computational domain is the periodic box $[-0.5, 0.5]^2$, and for Figures 5.7c-5.7e, the computational domain is the periodic box $[-4, 4]^2$. Ω is the region in the computational domain exterior to a circle of radius 0.25, centered at the origin. A boundary point spacing of $\Delta s \approx 0.75\Delta x$ is used, and solution values within $m_1 = 6$ meshwidths from the boundary are replaced using a second interpolation point $m_2 = 8$ meshwidths away from the boundary. All errors reported are absolute errors. Figure 5.7e gives a plot of the solution errors for Equation (5.74) for a mesh size of $N = 2^{12}$.

81

(a) Equation (5.74), $L = 8$ (b) Equation (5.75), $L = 8$
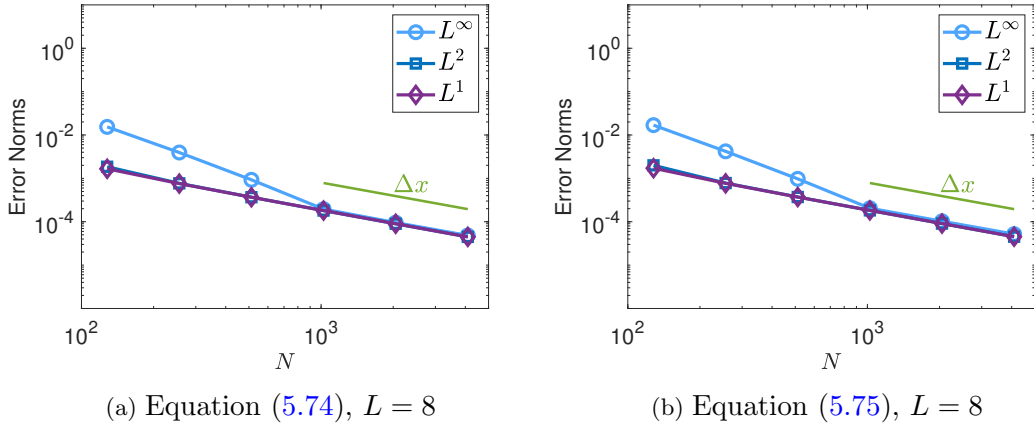
FIGURE 5.8. Refinement studies for solutions to Equations (5.74)-(5.75) found using the completed double layer method, with finite differences. The computational domain is the periodic box $[-4, 4]^2$. Ω is the region in the computational domain exterior to a circle of radius 0.25, centered at the origin. A boundary point spacing of $\Delta s \approx 0.75\Delta x$ is used, and solution values within $m_1 = 6$ meshwidths from the boundary are replaced using a second interpolation point $m_2 = 8$ meshwidths away from the boundary. All errors reported are absolute errors.

We next apply the completed IBDL method, described in Equation (5.48), with $\eta = 10$, to the two PDEs on the large computational domain of length $L = 8$. Figure 5.8 illustrates we get first order convergence for both PDEs with the completed method. However, we note that since we lose the second-order convergence for the PDE in Equation (5.75), the errors for that PDE are larger. Further investigations need to be made into the different results from these two boundary value problems.

5.5 Near-boundary convergence

As was discussed in Section 5.1.2, the discontinuity in the solution across the boundary prevents the method from achieving first-order convergence in the max norm near the boundary. As seen in Figure 5.2, the largest errors are confined to a region of about 2-3 meshwidths due to the support of the discrete delta function. However, there is also a wider region to which the numerical method spreads the errors. The physical length of the region on which the pointwise error fails to converge approaches 0 as the grid is refined, so if one only needs the solution away from the boundary, one can proceed with the method as illustrated, or even omit the interpolation step altogether.

82

(a) Finite difference, Equation (5.70) (b) Fourier spectral, Equation (5.70)
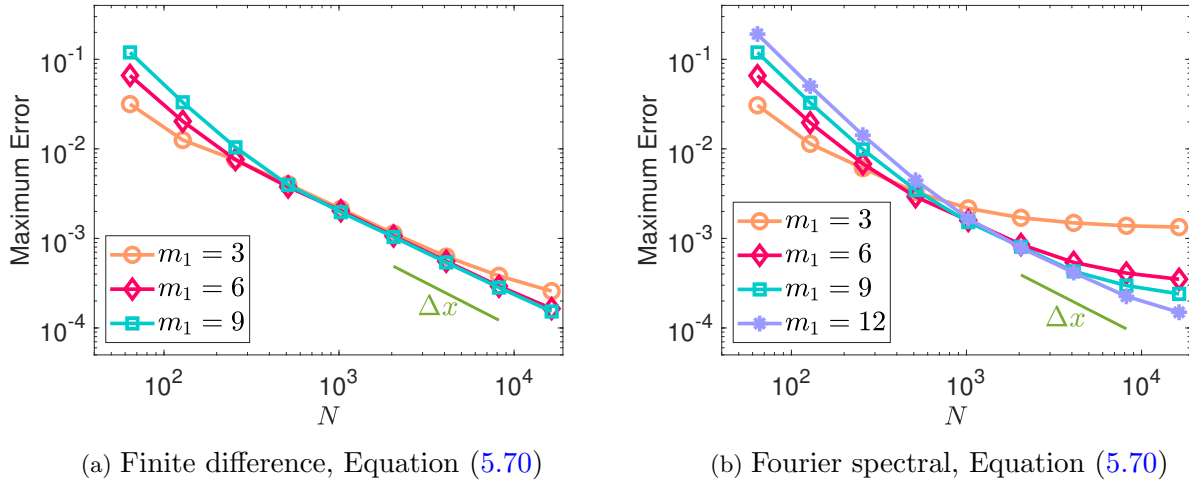
FIGURE 5.9. The L^∞ refinement studies for solving Equation (5.70) with the IBDL method. The computational domain is the periodic box $[-0.5, 0.5]^2$, Ω is the interior of a circle of radius 0.25, the boundary point spacing is $\Delta s \approx 0.75\Delta x$, and m_1 is varied. Figure 5.9a is found using a finite difference method for discretization of the PDE, and Figure 5.9b is found using a Fourier spectral method.

However, in this section, we will examine this issue more closely and investigate choices that can be made to recover pointwise convergence for the entire PDE domain.

5.5.1 Finite difference vs. Fourier spectral discretization

We found that one key factor affecting the spread of error is the numerical method used to discretize the PDE operator. We therefore start by revisiting Equation (5.70) and utilizing both finite difference and Fourier spectral methods for discretization. We extend our refinement study to an even finer mesh of $N = 2^{14}$ and explore different values for m_1, the number of meshwidths from the boundary for which interpolation is used. For all of these, we use an interior interpolation point that is $m_2 = m_1 + 2$ meshwidths from the boundary. Figure 5.9 shows the corresponding L^∞ refinement studies. In Figure 5.9a, we can see that $m_1 = 6$ is sufficient to maintain first-order pointwise convergence when using a finite difference discretization. Using a Fourier spectral method, on the other hand, requires a larger value of m_1. For low, fixed values of m_1, we see that once the error is small enough, the Fourier spectral method gives a diminishing rate of pointwise convergence and eventually a failure to converge.

To further explore this phenomenon, we next look at the PDE

$$\Delta u - u = -(x + y) \qquad \text{in } \Omega \qquad\qquad\qquad (5.76a)$$

$$u = x + y \qquad\qquad \text{on } \Gamma, \qquad\qquad\qquad (5.76b)$$

where Ω is the interior of a circle of radius 0.25, centered at the origin. The analytical solution is given by $u = x + y$. Figures 5.10a-5.10b show the L^∞ refinement studies for this problem. Due to the linearity of the solution, the primary source of error is the discontinuity, and this allows us to observe the aforementioned behavior on coarser grids. We can see that the lack of convergence with the Fourier spectral method is worse for this problem, but we see that $m_1 = 6$ is still sufficient to maintain convergence using a finite difference method.

To explore this phenomenon a little more closely, we also omit the interpolation step of the method and examine cross-sections of the error, located at $y = 0$, for which the boundary point is located at $x = -0.25$. Figures 5.10c-5.10d show these error cross-sections on a relatively coarse grid of $N = 2^8$ and a fine grid of $N = 2^{14}$. Each marker on the plot represents a grid point on the mesh. For the coarse grid, we see in Figure 5.10c that the pointwise errors for both methods decrease to the size of the interior error within a couple meshwidths. For the fine grid, however, the Fourier spectral method takes more meshwidths to decrease to the level of the interior error than the finite difference method. This illustrates the further spreading out of the error that takes place when using the Fourier spectral method due to the slow convergence of errors in a truncated Fourier series for a discontinuity. It is important to note, however, that while the number of meshwidths affected increases as the grid is refined, the physical distance affected still decreases. This is clear when observing the different scaling on the x-axes between Figures 5.10c and 5.10d.

In order to recover pointwise convergence when utilizing a Fourier spectral method, we can choose a function for m_1 that increases as the mesh is refined. We have found that increasing logarithmically seems sufficient for these problems. For example, one could simply increase m_1 by 2 each time the number of grid points, N, is doubled. By doing this, we use interpolation at a greater number of meshwidths as we refine the grid, but, again, the physical distance on which we use interpolation is still decreasing relatively quickly towards 0. Figures 5.11a and 5.11b illustrate using such a function for m_1 for Equations (5.70) and (5.76), respectively. The function we use is

84

(a) Finite difference, Equation (5.76)

(b) Fourier spectral, Equation (5.76)
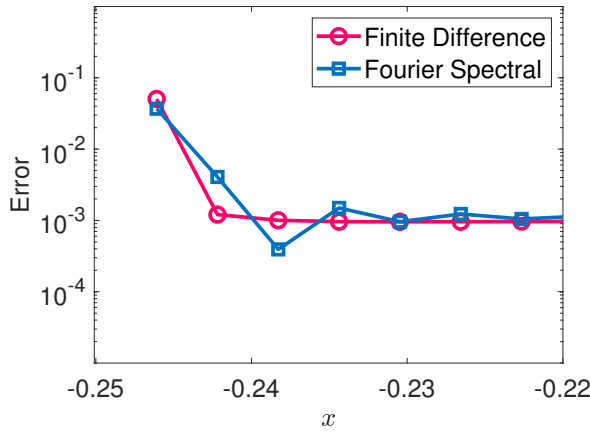
(c) Pointwise errors on coarse grid

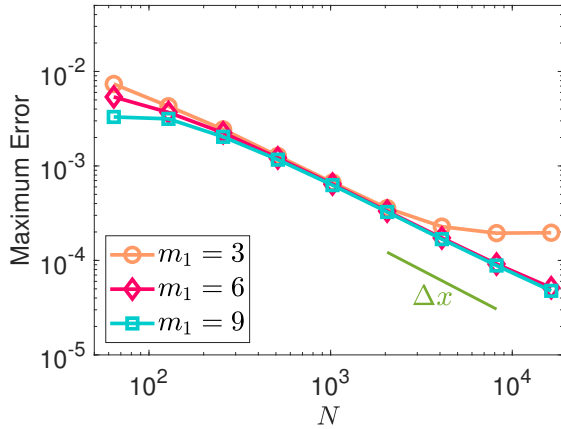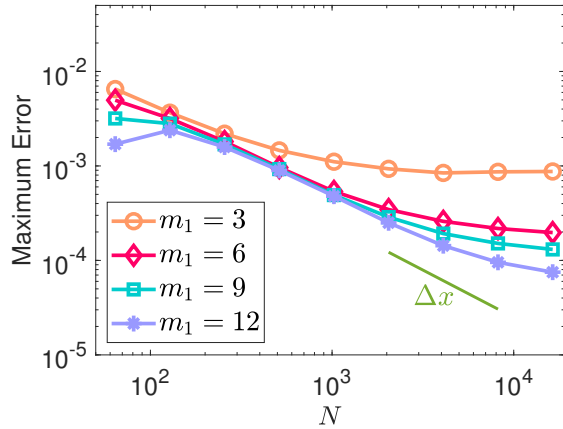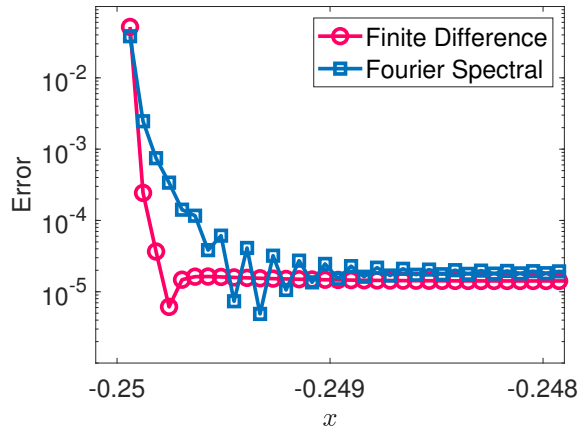(d) Pointwise errors on fine grid

FIGURE 5.10. Figures 5.10a-5.10b give the L^∞ refinement studies for solving Equation (5.76) with the IBDL method for varying values of m_1. The computational domain is the periodic box $[-0.5, 0.5]^2$, Ω is the interior of a circle of radius 0.25, and the boundary point spacing is $\Delta s \approx 0.75\Delta x$. Figure 5.10a is found using a finite difference method for discretization of the PDE, and Figure 5.10b is found using a Fourier spectral method. Figures 5.10c-5.10d show the errors along the line $y = 0$ when solving Equation (5.76) with the IBDL method with no interpolation and using both finite difference and Fourier spectral methods for discretization of the PDE. Figure 5.10c uses $N = 2^8$, and Figure 5.10d uses $N = 2^{14}$. Each plot marker represents a gridpoint on the corresponding mesh.

$m_1 = 2(\log_2 N - 4)$, which gives m_1 values ranging from 4 to 20 on these grid sizes. We see that this option recovers pointwise convergence.

85

(a) Fourier spectral, Equation (5.70)   (b) Fourier spectral, Equation (5.76)
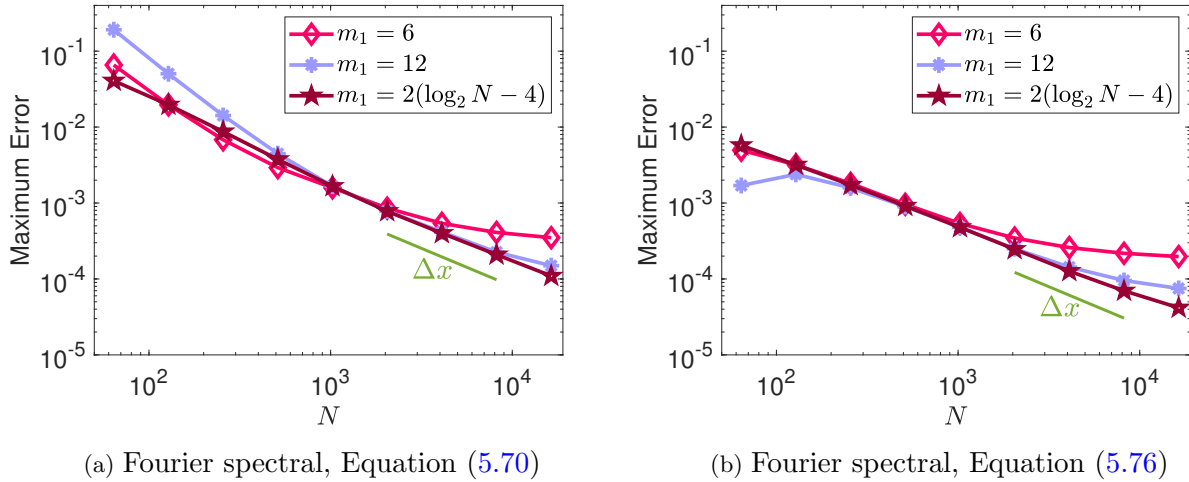
FIGURE 5.11. The L^∞ refinement studies for solving Equations (5.70) and (5.76) using a Fourier spectral method for discretization of the PDE. The value of m_1 is again varied, but also included is an increasing function for m_1 given by $m_1 = 2(\log_2{(N)} - 4)$.

5.5.2 Boundary point spacing

In addition to the PDE discretization and interpolation width, one other factor affecting the error spread is the boundary point spacing. We have observed that in general, using tightly spaced boundary points gives better pointwise convergence for a fixed interpolation width, especially in the case of the finite difference method. Specifically, we have seen that keeping $\Delta s \leq 0.75\Delta x$ will help to avoid a slowing pointwise convergence rate. This is illustrated in Figure 5.12a, which gives the L^∞ refinement studies for Equation (5.76) using a finite difference discretization, $m_1 = 6$, and various the boundary point spacings. Furthermore, since we saw in Sections 5.4.1 and 5.4.2 that the iteration count remains essentially unchanged by tightening the boundary points, the only numerical drawback to using this tighter point spacing is that each iteration is a little more expensive.

If one does want to use more widely spaced points, one can simply increase the interpolation widths suggested in the previous section. Figure 5.12b shows the L^∞ refinement studies for Equation (5.76) when the boundary point spacing is given by $\Delta s \approx 2\Delta x$ and the interpolation width is varied. In this case of more widely spaced boundary points, increasing m_1 logarithmically as the mesh is refined again recovers pointwise convergence.

86

(a) Fixed $m_1 = 6$ (b) Fixed $\Delta s \approx 2\Delta x$
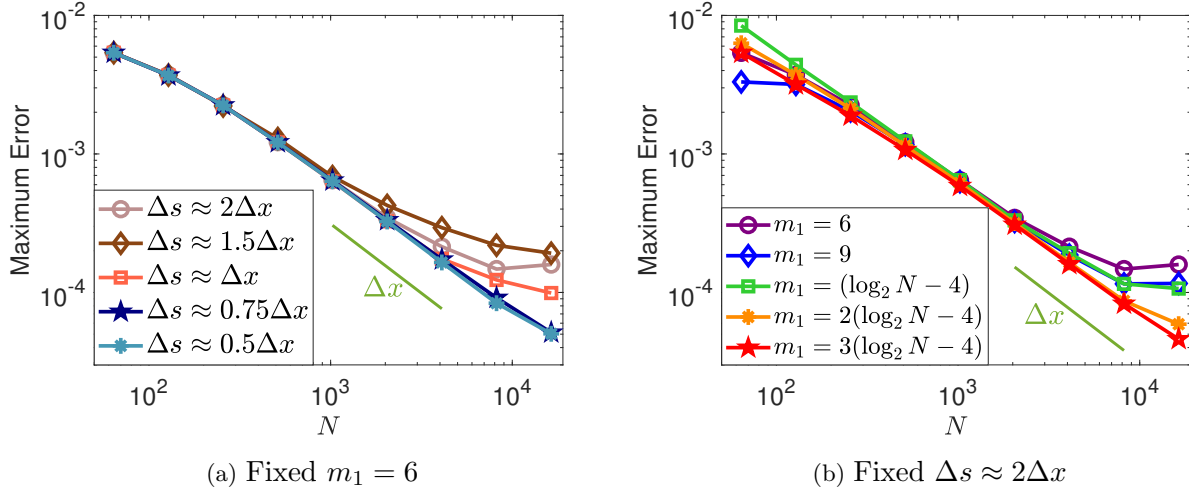
FIGURE 5.12. The L^∞ refinement studies for solving Equation (5.76) with the IBDL method using a finite difference discretization of the PDE. Figure 5.12a uses a fixed interpolation width of $m_1 = 6$ meshwidths and a second interpolation point $m_2 = 8$ meshwidths from the boundary, and the boundary point spacing is varied. Figure 5.12b has a wide boundary point spacing of $\Delta s \approx 2\Delta x$, and the interpolation width m_1 is varied.

In accordance with the observations seen in Section 5.5, for the majority of Chapter 5, we have elected to use a finite difference PDE discretization, a boundary point spacing of $\Delta s \approx 0.75\Delta x$, an interpolation width of $m_1 = 6$ meshwidths, and a second interpolation point $m_2 = 8$ meshwidths from the boundary. Similar results can be found by using a Fourier spectral method, a boundary point spacing of $\Delta s \approx 0.75\Delta x$, an interpolation width given by $m_1 = 2(\log_2(N) - 4)$, and a second point $m_2 = m_1 + 2$ meshwidths from the boundary. We reiterate that using these options is not essential if one is only concerned with the solution away from the boundary because the width of the region on which the error does not converge pointwise does approach 0 as the grid is refined. For reasons discussed in Sections 6.3.1 and 6.4.3, we primarily make use of Fourier spectral methods in Chapter 6.

5.6 Results: convergence of potential strength

In addition to the increased speed with which we solve the system for the potential strength, Q, another benefit of the better conditioning of the IBDL method is that we are more easily able get convergence in this potential strength. It is a well-known problem that the poor conditioning of the

Schur complement results in a noisy force distribution in the IBSL method [**21**], and this noise is larger for tighter boundary point spacing. Here, we illustrate that in the IBDL method, we obtain much smoother distributions with little need for filtering.

To demonstrate this, we compute the IBSL and IBDL force distributions, F and Q, obtained by solving the interior problem from Section 5.4.1, where the computational domain is instead the periodic box $[-10, 10]^2$. For estimating convergence, we approximate the exact distributions using a boundary element method, and this larger computational domain allows us to approximate the periodic Green's function with the free-space Green's function.

In Figures 5.13a-5.13d, for which we use $\Delta s \approx \Delta x$, we see approximately first-order convergence for the IBDL distribution Q and no convergence for the IBSL distribution F. Figure 5.13a shows that F quickly becomes very noisy as we refine the grid. As seen in Figure 5.13b, there is much less noise in Q. Note that the plots only show a portion of the distributions in order to more easily view this noise.

For Figures 5.13e-5.13h, we increase the spacing for the boundary points to $\Delta s \approx 2\Delta x$. Then we see some convergence in the IBSL distribution F. However, the IBDL potential strength Q achieves greater convergence and a much lower relative error than F. Therefore, we see that the IBDL method in general allows us to achieve a much more accurate distribution, and the next section demonstrates an application for which this convergence is vital.

5.7   Results: Neumann boundary conditions

As discussed in Section 5.2, we are able to use the connection to integral equations to solve a PDE with Neumann boundary conditions by solving

$$(-S^*\mathcal{L}^{-1}\widetilde{S})U_b - \frac{1}{2}U_b = S^*\mathcal{L}^{-1}SV_b - S^*\mathcal{L}^{-1}\tilde{g} \tag{5.77}$$

for the unknown boundary values U_b. By doing this, we are again inverting an operator corresponding to a second-kind integral operator with a small condition number. Therefore, we are able to get convergence in this boundary distribution, as discussed in the previous section. When we have Neumann boundary conditions, the convergence of the distribution, U_b is especially important

(a) IBSL, $\Delta s \approx \Delta x$

(b) IBDL, $\Delta s \approx \Delta x$

(c) IBSL, $\Delta s \approx \Delta x$

(d) IBDL, $\Delta s \approx \Delta x$

(e) IBSL, $\Delta s \approx 2\Delta x$

(f) IBDL, $\Delta s \approx 2\Delta x$

(g) IBSL, $\Delta s \approx 2\Delta x$

(h) IBDL, $\Delta s \approx 2\Delta x$
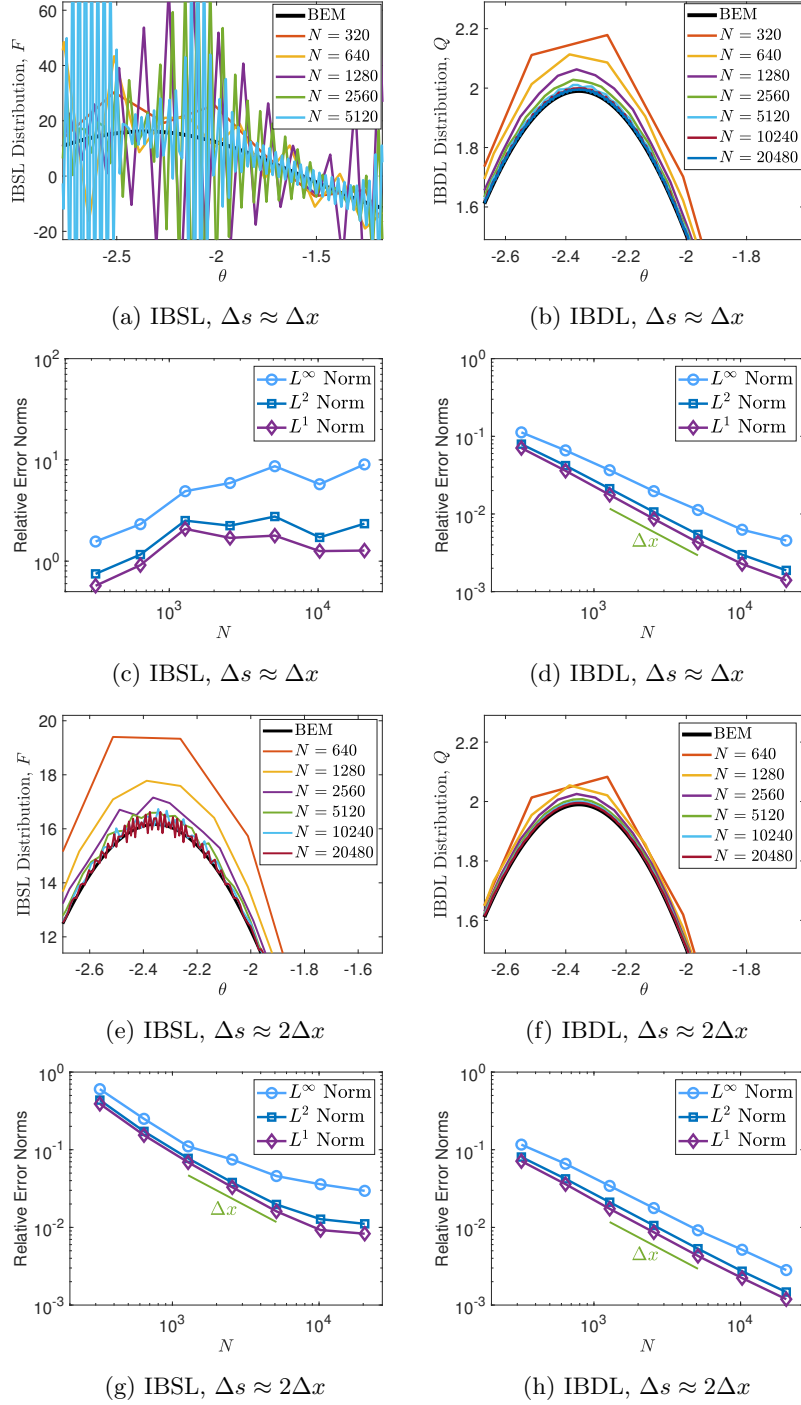
FIGURE 5.13. Figures 5.13a-5.13b and 5.13e-5.13f show portions of the IBSL and IBDL potential strengths for Equation (5.70), plotted against the angle θ. Boundary point spacing is given by $\Delta s \approx \Delta x$ for Figures 5.13a-5.13b and $\Delta s \approx 2\Delta x$ for Figures 5.13e-5.13f. Black curves correspond to distributions found using a boundary element method with uniform, straight line elements and Gauss-Legendre quadrature. The BEM distributions are used to approximate the error, normalized by the maximum value. Figures 5.13c-5.13d and 5.13g-5.13h show the corresponding refinement studies.

89

(a) Solution refinement study

(b) U_b refinement study
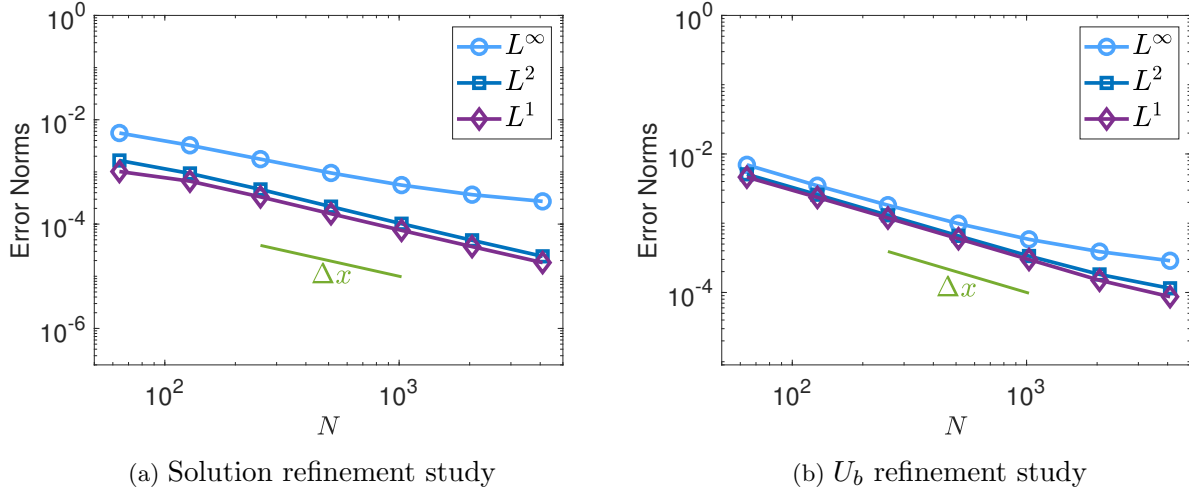
FIGURE 5.14. Refinement studies for solving Equation (5.78) with the method discussed in Section 5.2. The computational domain is the periodic box $[-0.5, 0.5]$, Ω is the interior of a circle of radius 0.25, and the prescribed normal derivatives on the boundary are given by $\partial u/\partial n|_\Gamma = 8(x^2 - y^2)$. We use a boundary point spacing of $\Delta s \approx 0.75\Delta x$ and replace solution values within $m_1 = 6$ meshwidths from the boundary using interior interpolation points $m_2 = 8$ meshwidths away from the boundary. Figures 5.14a and 5.14b show the refinement studies for the absolute solution errors and the boundary values, U_b, respectively.

because U_b is used when performing the interpolation step for grid points near the boundary. In the Dirichlet case, U_b is known, providing us with more accurate interpolation data, but in the case of Neumann, the values both in the interior and on the boundary are approximate.

We consider the PDE given by

$$\Delta u - u = -(x^2 - y^2) \text{ in } \Omega \tag{5.78a}$$

$$\frac{\partial u}{\partial n} = 8(x^2 - y^2) \qquad \text{on } \Gamma, \tag{5.78b}$$

where Ω is the interior of a circle of radius 0.25, centered at the origin. The analytical solution is given by $u = x^2 - y^2$. Our computational domain here is the periodic box $[-0.5, 0.5]^2$, and we use a finite difference method. Our boundary point spacing is given by $\Delta s \approx 0.75\Delta x$, and we replace solution values within $m_1 = 6$ meshwidths of the boundary using interior interpolation points located $m_2 = 8$ meshwidths from the boundary. For this problem, our method only requires $4 - 6$ iterations of `gmres`. Figure 5.14 illustrates the refinement studies for the solution values and U_b, and we see approximately first-order convergence in both of these.

Chapter 6

The Immersed Boundary Double Layer (IBDL) method: Brinkman, Stokes and Navier-Stokes equations

In this chapter, we present the Immersed Boundary Double Layer method for problems governing fluid flow, including Stokes, Brinkman, and Navier-Stokes equations. As seen in Section 2.3, the IBSL method can be adapted to the Stokes equation quite simply. The term that introduces the required boundary density, $S\boldsymbol{F}$, remains unchanged. However, adapting the IBDL method to Stokes equation is nontrivial, and we must use the form of the Green's function stress tensor, $\boldsymbol{T}$, to establish the required form.

This chapter is organized as follows. In Section 6.1, we introduce the method and demonstrate the connection to a regularized double layer integral equation for Stokes flow. We also modify the method for an exterior domain in order to represent flows with net force and torque. In Section 6.2, we present the method as applied to a fluid equation with Neumann boundary conditions. In Section 6.3, we discuss the numerical implementation, focusing on aspects that differ from the previous IBDL method for scalar elliptic problems. In Section 6.4, we present numerical results for applications to Stokes and Brinkman equations, and in Section 6.5, we present numerical results for the Navier-Stokes equation.

6.1 Mathematical description of the method

We now look at the boundary value problem given by

$$\mathcal{L}\boldsymbol{u} - \nabla p = \boldsymbol{g} \qquad \text{in } \Omega \tag{6.1a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{6.1b}$$

$$\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma, \tag{6.1c}$$

where $\mathcal{L} = \mu\Delta - k^2$, $\boldsymbol{u}$ is the two-dimensional velocity, and p is the pressure. We again assume that Ω is embedded in a larger computational domain $\mathcal{C}$ and that the boundary of Ω is given by Γ, and it is a smooth one-dimensional curve parametrized by $\boldsymbol{X}(s)$.

6.1.1 Formulation and motivation

The Immersed Boundary Double Layer formulation for Equation (6.1) is given by

$$\mathcal{L}\boldsymbol{u} - \nabla p + \mu\nabla \cdot (S\boldsymbol{A}) = \tilde{\boldsymbol{g}} \qquad \text{in } \mathcal{C} \tag{6.2a}$$

$$\nabla \cdot \boldsymbol{u} + S(\boldsymbol{Q} \cdot \boldsymbol{n}) = 0 \qquad \text{in } \mathcal{C} \tag{6.2b}$$

$$S^*\boldsymbol{u} + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b \qquad \text{on } \Gamma, \tag{6.2c}$$

where we define $\boldsymbol{A}$ as

$$\boldsymbol{A} \equiv \boldsymbol{Q} \otimes \boldsymbol{n} + \boldsymbol{n} \otimes \boldsymbol{Q}, \tag{6.3}$$

or in Einstein notation,

$$A_{ij} = Q_i n_j + Q_j n_i. \tag{6.4}$$

We again assume an arclength parametrization of Γ, and $\boldsymbol{n}$ is the unit normal pointing out of Ω. Here, $\boldsymbol{Q}$ is a vector-valued distribution supported on Γ, and it plays the role of the Lagrange multiplier enforcing the boundary condition. The term given by $\nabla \cdot (S\boldsymbol{A})$ can be seen as a generalization

of the term $\widetilde{S}Q \equiv \nabla \cdot (SQ\boldsymbol{n})$ in the IBDL method for the Poisson and Helmholtz equations. Also note the presence of an additional term in Equation (6.2b) given by $S(\boldsymbol{Q} \cdot \boldsymbol{n})$. Lastly, recall that $\tilde{\boldsymbol{g}}$ represents an extension of the function $\boldsymbol{g}$ from Ω to the computational domain $\mathcal{C}$, and we define it as $\tilde{\boldsymbol{g}} \equiv \boldsymbol{g}\chi_\Omega + \boldsymbol{g}_e\chi_{\mathcal{C}\backslash\Omega}$.

We now present motivation for the form of this system of equations, and in Section 6.1.2, we explicitly verify the connection between the IBDL method and a regularized double layer integral equation. To motivate the structure of this system, we look again to the double layer integral representation presented in Section 3.5.3, given by

$$u_j(\boldsymbol{x}_0) = \int_\Gamma \Psi_i(\boldsymbol{x})T_{ijk}(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}). \tag{6.5}$$

If we decompose $\boldsymbol{T}$ into its constituents, we get

$$u_j(\boldsymbol{x}_0) = \int_\Gamma \Psi_i(\boldsymbol{x})\Big(\frac{\partial G_{ij}}{\partial \boldsymbol{x}_k}(\boldsymbol{x}, \boldsymbol{x}_0) + \frac{\partial G_{kj}}{\partial \boldsymbol{x}_i}(\boldsymbol{x}, \boldsymbol{x}_0)\Big)n_k(\boldsymbol{x})dl(\boldsymbol{x})$$
$$- \int_\Gamma \Psi_i(\boldsymbol{x})\delta_{ik}p_j(\boldsymbol{x}, \boldsymbol{x}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}), \quad (6.6)$$

where $\boldsymbol{G}$ is the Stokes Green's function and $\boldsymbol{p}$ is the corresponding pressure vector. This can also be rewritten as

$$u_j(\boldsymbol{x}_0) = \int_\Gamma \frac{\partial G_{ij}}{\partial \boldsymbol{x}_k}(\boldsymbol{x}, \boldsymbol{x}_0)\Big(\Psi_i(\boldsymbol{x})n_k(\boldsymbol{x}) + \Psi_k(\boldsymbol{x})n_i(\boldsymbol{x})\Big)dl(\boldsymbol{x})$$
$$- \int_\Gamma p_j(\boldsymbol{x}, \boldsymbol{x}_0)\Psi_k n_k(\boldsymbol{x})dl(\boldsymbol{x}). \quad (6.7)$$

The first term represents a symmetric distribution of point force dipoles, and this corresponds to the IBDL term in Equation (6.2a). Additionally, as in Chapter 5, we can associate the potential strength Ψ with the IBDL distribution $\boldsymbol{Q}$. By doing this, we can see that the strength of the point force dipoles matches our definition of $\boldsymbol{A}$. The second term corresponds to a distribution of point sources, and this explains the need for the extra source term in Equation (6.2b), and again the strength of this source matches our use of $\boldsymbol{Q} \cdot \boldsymbol{n}$.

6.1.2 Explicit connection between Stokes IBDL and boundary integral equation

We will now demonstrate the connection by focusing on the homogeneous Immersed Boundary Double Layer system for Stokes equation, and for simplicity, we assume the invertibility of Δ. The homogeneous IBDL system for Stokes is

$$\mu\Delta\boldsymbol{u} - \nabla p + \mu\nabla\cdot(S\boldsymbol{A}) = 0 \qquad \text{in } \mathcal{C} \tag{6.8a}$$

$$\nabla\cdot\boldsymbol{u} + S(\boldsymbol{Q}\cdot\boldsymbol{n}) = 0 \qquad\qquad \text{in } \mathcal{C} \tag{6.8b}$$

$$S^*\boldsymbol{u} + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b \qquad\qquad \text{on } \Gamma. \tag{6.8c}$$

Taking the divergence of Equation (6.8a) and using Equation (6.8b), we get

$$\Delta p = \mu\nabla\cdot\nabla\cdot(S\boldsymbol{A}) + \mu\Delta(\nabla\cdot\boldsymbol{u}) = \mu\nabla\cdot\nabla\cdot(S\boldsymbol{A}) - \mu\Delta(S(\boldsymbol{Q}\cdot\boldsymbol{n})). \tag{6.9}$$

Using the definition of the spread operator, this becomes

$$\Delta p = \mu\nabla\cdot\nabla\cdot\int_\Gamma \boldsymbol{A}(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds - \mu\Delta\int_\Gamma \boldsymbol{Q}(s)\cdot\boldsymbol{n}(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{6.10}$$

We can then invert the operator and use the property of the regularized Green's function for Laplace's equation, $\Delta G_h^L(\boldsymbol{x}, \boldsymbol{x}_0) = -\delta_h(\boldsymbol{x} - \boldsymbol{x}_0)$. We then have

$$p(\boldsymbol{x}) = -\mu\nabla\cdot\nabla\cdot\int_\Gamma \boldsymbol{A}(s)G_h^L(\boldsymbol{x}, \boldsymbol{X}(s))ds + \mu\Delta\int_\Gamma \boldsymbol{Q}(s)\cdot\boldsymbol{n}(s)G_h^L(\boldsymbol{x}, \boldsymbol{X}(s))ds. \tag{6.11}$$

For clarity, we will now switch to Einstein notation. Bringing the operators into the integrals and manipulating the expressions, we get

$$p(\boldsymbol{x}) = -\mu\int_\Gamma \frac{\partial^2 G_h^L}{\partial x_i \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s))A_{ik}(s)ds + \mu\int_\Gamma \frac{\partial^2 G_h^L}{\partial x_m \partial x_m}(\boldsymbol{x}, \boldsymbol{X}(s))Q_k(s)n_k(s)ds. \tag{6.12}$$

Using this expression for pressure in Equation (6.8a) and canceling the μ on all terms, we get

$$\Delta u_j = -\int_\Gamma \frac{\partial^3 G_h^L}{\partial x_i \partial x_j \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s))A_{ik}(s)ds + \int_\Gamma \frac{\partial^3 G_h^L}{\partial x_m \partial x_m \partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s))Q_k(s)n_k(s)ds$$
$$- \frac{\partial}{\partial x_k}\int_\Gamma A_{jk}(s)\delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds. \tag{6.13}$$

94

We next invert the operator. For the first term, we use that $\Delta H_h = G_h^L$ from Equation (4.17). For the second term, we simply cancel the Δ in the integral, and for the third term, we use $\Delta^2 H_h = -\delta_h$ from Equation (4.15). We then get

$$u_j(\boldsymbol{x}) = -\int_\Gamma \frac{\partial^3 H_h}{\partial x_i \partial x_j \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{ik}(s) ds + \int_\Gamma \frac{\partial G_h^L}{\partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s) ds$$
$$+ \int_\Gamma \frac{\partial^3 H_h}{\partial x_m \partial x_m \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{jk}(s) ds. \quad (6.14)$$

Switching the last two integrals and rearranging, we get

$$u_j(\boldsymbol{x}) = \int_\Gamma \frac{\partial}{\partial x_k}\left(-\frac{\partial^2 H_h}{\partial x_i \partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{ik}(s) + \frac{\partial^2 H_h}{\partial x_m \partial x_m}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{jk}(s)\right) ds$$
$$+ \int_\Gamma \frac{\partial G_h^L}{\partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s) ds. \quad (6.15)$$

We can also write this as

$$u_j(\boldsymbol{x}) = \int_\Gamma \frac{\partial}{\partial x_k}\left(-\frac{\partial^2 H_h}{\partial x_i \partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) + \delta_{ij}\frac{\partial^2 H_h}{\partial x_m \partial x_m}(\boldsymbol{x}, \boldsymbol{X}(s))\right) A_{ik}(s) ds$$
$$+ \int_\Gamma \frac{\partial G_h^L}{\partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s) ds. \quad (6.16)$$

Recognizing the form of the regularized Stokes Green's function from Equation (4.30), we have

$$u_j(\boldsymbol{x}) = \int_\Gamma \frac{\partial G_{ij}^h}{\partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{ik}(s) ds + \int_\Gamma \frac{\partial G_h^L}{\partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s) ds. \quad (6.17)$$

Lastly, we can use our definition of $\boldsymbol{A}$ to write this as

$$u_j(\boldsymbol{x}) = \int_\Gamma \frac{\partial G_{ij}^h}{\partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s))\Big(Q_i(s) n_k(s) + Q_k(s) n_i(s)\Big) ds + \int_\Gamma \frac{\partial G_h^L}{\partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s) ds.$$
$$(6.18)$$

Using the the identity for the pressure vector given in Equation (4.31), this is

$$u_j(\boldsymbol{x}) = \int_\Gamma \frac{\partial G_{ij}^h}{\partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s))\Big(Q_i(s) n_k(s) + Q_k(s) n_i(s)\Big) ds - \int_\Gamma p_j^h(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s) ds.$$
$$(6.19)$$

95

Then, we recognize that by linearity, the decomposition of the stress tensor used in Equation (6.5) also holds using the regularized functions. Then by reversing the process we took to decompose $\boldsymbol{T}$ in Equations (6.5)-(6.7), we get

$$u_j(\boldsymbol{x}) = \int_\Gamma Q_i(s)T_{ijk}^h(\boldsymbol{x}, \boldsymbol{X}(s))n_k(s)ds, \tag{6.20}$$

for $\boldsymbol{x} \in \Omega \setminus \Gamma$. Next, we use Equation (6.8c) to get

$$\boldsymbol{U}_b(s') = \int_\mathcal{C} \boldsymbol{u}(\boldsymbol{x})\delta_h(\boldsymbol{x} - \boldsymbol{X}(s'))d\boldsymbol{x} + \frac{1}{2}Q_j(s'). \tag{6.21}$$

We then follow the same steps seen in Section 4.1.2. We combine this with Equation (6.20), change the order of integration, recognize the presence of $T_{ijk}^h * \delta_h$, and denote the twice-regularized stress tensor by T_{ijk}^{hh}. This leaves us with

$$U_{b_j}(s') = \int_\Gamma Q_i(s)T_{ijk}^{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s))n_k(s)ds + \frac{1}{2}Q_j(s'). \tag{6.22}$$

The odd symmetry of the stress tensor, which is preserved through convolutions with the regularized delta function, gives us that $T_{ijk}^{hh}(\boldsymbol{X}(s'), \boldsymbol{X}(s)) = -T_{ijk}^{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))$ [50]. Using this to switch the arguments of T_{ijk} and appropriately redefining $\boldsymbol{Q}$ and $\boldsymbol{U}_b$ as functions of $\boldsymbol{x}$, we get

$$U_{b_j}(\boldsymbol{X}(s')) = -\int_\Gamma Q_i(\boldsymbol{X}(s))T_{ijk}^{hh}(\boldsymbol{X}(s), \boldsymbol{X}(s'))n_k(\boldsymbol{X}(s))ds + \frac{1}{2}Q_j(\boldsymbol{X}(s')). \tag{6.23}$$

Recall that the double layer integral equation for $\boldsymbol{X}_0$ on the boundary from Section 3.6 is given by

$$U_{b_j}(\boldsymbol{X}_0) = \int_\Gamma^{PV} \Psi_i(\boldsymbol{x})T_{ijk}(\boldsymbol{x}, \boldsymbol{X}_0)n_k(\boldsymbol{x})dl(\boldsymbol{x}) - \frac{1}{2}\Psi_j(\boldsymbol{X}_0). \tag{6.24}$$

Using our arclength parametrization, we can write this as

$$U_{b_j}(\boldsymbol{X}(s')) = \int_\Gamma^{PV} \Psi_i(\boldsymbol{X}(s))T_{ijk}(\boldsymbol{X}(s), \boldsymbol{X}(s'))n_k(\boldsymbol{X}(s))ds - \frac{1}{2}\Psi_j(\boldsymbol{X}(s')). \tag{6.25}$$

By comparing Equations (6.23) and (6.25), we can see that in the limit that the regularization width, h, approaches 0, we have

$$\boldsymbol{Q}(\boldsymbol{X}) = -\boldsymbol{\Psi}(\boldsymbol{X}). \tag{6.26}$$

96

We then see from Section 3.5.3 that $\boldsymbol{Q}$ gives the jump in solution values across the boundary. We have now established that the IBDL method for Stokes is equivalent to a *regularized* double layer integral equation.

6.1.3 Exterior domains

As discussed in Section 3.5.3, the double layer potential is only capable of representing a flow with no net force or net torque. Therefore, for exterior domains, we will use the completed IBDL method, which was presented for the Poisson case in Section 5.3.3. Specifically, we add a single layer potential whose strength is given by a constant multiple of the double layer potential strength. For Stokes and Brinkman equations, the formulation is

$$\mathcal{L}\boldsymbol{u} - \nabla p + \eta S\boldsymbol{Q} + \mu\nabla \cdot (S\boldsymbol{A}) = \tilde{\boldsymbol{g}} \qquad \text{in } \mathcal{C} \tag{6.27a}$$

$$\nabla \cdot \boldsymbol{u} + S(\boldsymbol{Q} \cdot \boldsymbol{n}) = 0 \qquad \text{in } \mathcal{C} \tag{6.27b}$$

$$S^*\boldsymbol{u} + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b \qquad \text{on } \Gamma \tag{6.27c}$$

where $\mathcal{L} = \mu\Delta - k^2$ and $\boldsymbol{A}$ is defined by

$$A_{ij} = Q_i n_j + Q_j n_i. \tag{6.28}$$

Additionally, η is a positive constant, and in Section 6.4.2, we explore the affects of different choices for η. This supplementary flow is only absolutely required in the case of the Stokes equation, or $k = 0$. However, for the Brinkman equation, as k approaches 0, including this term becomes important for maintaining low errors and first-order convergence. However, as is discussed further in Section 6.4.2, including the term can lower errors even for k away from 0.

Force and Torque. We now derive the equations for net force and net torque corresponding to the completed IBDL method. If we start from Equation (6.27) and use the derivation in the previous section, the expression for pressure given in Equation (6.12) would have an additional

term resulting from the added single layer potential. We would then have

$$p(\boldsymbol{x}) = -\mu \int_\Gamma \frac{\partial^2 G_h^L}{\partial x_i \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{ik}(s)ds + \mu \int_\Gamma \frac{\partial^2 G_h^L}{\partial x_m \partial x_m}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s)ds$$

$$- \eta \int_\Gamma \frac{\partial G_h^L}{\partial x_i}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_i(s)ds. \quad (6.29)$$

Using this expression for pressure in Equation (6.27a) and dividing by μ, we get

$$\Delta u_j = - \int_\Gamma \frac{\partial^3 G_h^L}{\partial x_i \partial x_j \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{ik}(s)ds + \int_\Gamma \frac{\partial^3 G_h^L}{\partial x_m \partial x_m \partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s)ds$$

$$- \frac{\partial}{\partial x_k} \int_\Gamma A_{jk}(s) \delta_h(\boldsymbol{x} - \boldsymbol{X}(s))ds$$

$$- \frac{\eta}{\mu} \int_\Gamma \frac{\partial^2 G_h^L}{\partial x_i \partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_i(s)ds - \frac{\eta}{\mu} \int_\Gamma Q_j(s) \delta_h(\boldsymbol{x}, \boldsymbol{X}(s))ds, \quad (6.30)$$

where the last two integrals are the contribution from the single layer term. We next invert the operator and use the same identities used in the previous section for the first three terms. For the last two terms, we use the identities used in Section 4.2.2 to go from Equation (4.37) to Equation (4.38). Then we get

$$u_j(\boldsymbol{x}) = - \int_\Gamma \frac{\partial^3 H_h}{\partial x_i \partial x_j \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{ik}(s)ds + \int_\Gamma \frac{\partial G_h^L}{\partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_k(s) n_k(s)ds$$

$$+ \int_\Gamma \frac{\partial^3 H_h}{\partial x_m \partial x_m \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s)) A_{jk}(s)ds$$

$$- \frac{\eta}{\mu} \int_\Gamma \frac{\partial^2 H_h}{\partial x_i \partial x_j}(\boldsymbol{x}, \boldsymbol{X}(s)) Q_i(s)ds + \frac{\eta}{\mu} \int_\Gamma Q_j(s) \frac{\partial^2 H_h}{\partial x_k \partial x_k}(\boldsymbol{x}, \boldsymbol{X}(s))ds. \quad (6.31)$$

From the previous section, we know that the first three integrals simplify to the double layer potential. Then, from Section 4.2.2, we know that the last two integrals simplify to a single layer potential. We therefore have

$$u_j(\boldsymbol{x}) = \int_\Gamma Q_i(s) T_{ijk}^h(\boldsymbol{x}, \boldsymbol{X}(s)) n_k(s)ds + \frac{\eta}{\mu} \int_\Gamma G_{ij}^h(\boldsymbol{x}, \boldsymbol{X}(s)) Q_i(s)ds. \quad (6.32)$$

Specifically, we see that with the formulation in Equation (6.27), we retrieve the integral representation from Equation (3.68), with the correct constant on the second term. Using that $\boldsymbol{Q} = -\boldsymbol{\Psi}$, we therefore know from Section 3.5.3 that the net force and net torque through a curve enclosing

98

Γ is given by

$$\boldsymbol{B} = -\int_{\Gamma} \boldsymbol{Q}(s)ds \tag{6.33a}$$

$$L = -\int_{\Gamma} \boldsymbol{X}(s) \times \boldsymbol{Q}(s)ds, \tag{6.33b}$$

and we will use this for calculating net force in Section 6.4.

6.2 Neumann boundary conditions

We now look at the PDE with Neumann boundary conditions given by

$$\mathcal{L}\boldsymbol{u} - \nabla p = \boldsymbol{g} \qquad \text{in } \Omega \tag{6.34a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{6.34b}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{F}_b \qquad \text{on } \Gamma, \tag{6.34c}$$

where $\mathcal{L} = \mu\Delta - k^2$. Here, we present the IBDL formulation for this PDE. For brevity, we omit the explicit connection to boundary integral equations. The steps follow closely with those from Sections 5.2, 6.1.2, and 6.1.3. The formulation is

$$\mathcal{L}\boldsymbol{u} - \nabla p + S(\boldsymbol{F}_b) + \mu\nabla \cdot (S\boldsymbol{A}) = \tilde{\boldsymbol{g}} \qquad \text{in } \mathcal{C} \tag{6.35a}$$

$$\nabla \cdot \boldsymbol{u} + S(\boldsymbol{U}_b \cdot \boldsymbol{n}) = 0 \qquad \text{in } \mathcal{C} \tag{6.35b}$$

$$S^*\boldsymbol{u} = \frac{1}{2}\boldsymbol{U}_b \qquad \text{on } \Gamma, \tag{6.35c}$$

where $\boldsymbol{A}$ is defined by

$$A_{ij} = U_{b_i}n_j + U_{b_j}n_i. \tag{6.36}$$

Here, the unknown potential strength that we solve for is the distribution of boundary values, $\boldsymbol{U}_b$.

6.3 Numerical implementation

There is no change in the discretization of the spread and interpolation operators discussed in Section 5.3. We also continue to use the same linear interpolation for solution values near the grid. In this section, we discuss the discretization of space and then the methods for solving the discrete Brinkman and Stokes equations.

6.3.1 Discretization of differential operators

As discussed in Section 2.3.1, taking a discrete divergence of a discrete gradient results in a wide Laplacian. In the case of the IBSL method, we have the option to avoid this issue by using a staggered grid, in which vector quantities are defined on the centers of the cell edges and scalar quantities are defined in the centers of the cells. In the IBSL method, this requires that the spread action, $S\boldsymbol{F}$, consists of spreading the F_1 components to vertical edge centers and the F_2 components to horizontal edge centers. The discrete divergence of $S\boldsymbol{F}$, which is used in the solution of pressure, then maps the $\nabla \cdot (S\boldsymbol{F})$ quantities to the cell centers. In the IBDL method, however, one must spread a *matrix*, $\boldsymbol{A}$. Using the definition of $\boldsymbol{A}$, we could also say we need to spread the vectors $\boldsymbol{Q}$ and $\boldsymbol{n}$, but we also need to combine elements on different edges, for instance, when we calculate $Q_1 n_2$. Therefore, using a staggered grid for the IBDL method requires a nontrivial adaptation, and this is an area for future research.

In the IBSL method, we can also use the wide Laplacian directly to solve for pressure, but in the case of the IBDL method, we fail to get first-order convergence with this approach. We conjecture that the interaction of two different differential operators results in inconsistent numerical Green's functions, and in the case of the IBDL method, this may lead to larger numerical errors due to the high frequency oscillations that would result from taking the derivative of the regularized delta function. This is again another area for future research. We illustrate this issue in Section 6.4.3, and we demonstrate that we are able to recover first order convergence by approximating the wide Laplacian with the usual 5-point second-order Laplacian and utilizing a smoother delta function. When we use finite differences for the IBDL method for Stokes, we therefore use the usual 5-point

Laplacian and the 6-point B-spline delta function utilized in [**4**]. It is given by

$$\delta_h = \frac{1}{h^2} \phi\left(\frac{x}{h}\right) \phi\left(\frac{y}{h}\right), \tag{6.37}$$

for $h = \Delta x = \Delta y$ and

$$\phi(r) = \begin{cases} \frac{11}{20} - \frac{1}{2}r^2 + \frac{1}{4}r^4 - \frac{1}{12}r^5 & 0 \le |r| \le 1 \\[2mm] \frac{17}{40} + \frac{5}{8}r - \frac{7}{4}r^2 + \frac{5}{4}r^3 - \frac{3}{8}r^4 + \frac{1}{24}r^5 & 1 \le |r| \le 2 \\[2mm] \frac{81}{40} - \frac{27}{8}r + \frac{9}{4}r^2 - \frac{3}{4}r^3 + \frac{1}{8}r^4 - \frac{1}{120}r^5 & 2 \le |r| \le 3 \\[2mm] 0 & |r| \ge 3. \end{cases} \tag{6.38}$$

However, we primarily utilize a Fourier spectral discretization for the IBDL method applied to Stokes, and this discretization is described in Section 2.3.1.

6.3.2   Solution to discrete system for Brinkman equation

Recall that the IBDL formulation of the PDE in Equation (6.1) is given by

$$\mathcal{L}\boldsymbol{u} - \nabla p + \eta S\boldsymbol{Q} + \mu \nabla \cdot (S\boldsymbol{A}) = \tilde{\boldsymbol{g}} \qquad \text{in } \mathcal{C} \tag{6.39a}$$

$$\nabla \cdot \boldsymbol{u} + S(\boldsymbol{Q} \cdot \boldsymbol{n}) = 0 \qquad \text{in } \mathcal{C} \tag{6.39b}$$

$$S^*\boldsymbol{u} + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b \qquad \text{on } \Gamma, \tag{6.39c}$$

where $\eta \ge 0$, and for this section we focus on $k > 0$. We may take $\eta = 0$ in the case of an interior domain or large k.

Taking the divergence of Equation (6.39a) and using Equation (6.39b), we get

$$\Delta p = -\mathcal{L}S(\boldsymbol{Q} \cdot \boldsymbol{n}) + \eta \nabla \cdot (S\boldsymbol{Q}) + \mu \nabla \cdot \nabla \cdot (S\boldsymbol{A}) - \nabla \cdot \tilde{\boldsymbol{g}}. \tag{6.40}$$

Equation (6.40) is solvable on the periodic domain, and we invert the periodic Laplacian with the operator Δ_0^{-1}, which was described in Section 2.2.4. We therefore fix its mean value on $\mathcal{C}$ to be 0.

Completing this inversion, we get

$$p = -\Delta_0^{-1}\mathcal{L}S(\boldsymbol{Q}\cdot\boldsymbol{n}) - \Delta_0^{-1}\,\nabla\cdot\Big(\tilde{\boldsymbol{g}} - \eta S\boldsymbol{Q} - \mu\nabla\cdot(S\boldsymbol{A})\Big). \tag{6.41}$$

Let us again denote the operator that projects onto divergence-free fields as $\mathbb{P} = \mathbb{I} - \nabla\Delta_0^{-1}\nabla\cdot$. Using this operator and the expression for p from Equation (6.41), Equation (6.39a) gives us

$$\mathcal{L}\boldsymbol{u} = -\nabla\Delta_0^{-1}\mathcal{L}S(\boldsymbol{Q}\cdot\boldsymbol{n}) + \mathbb{P}\Big(\tilde{\boldsymbol{g}} - \eta S\boldsymbol{Q} - \mu\nabla\cdot(S\boldsymbol{A})\Big). \tag{6.42}$$

Inverting $\mathcal{L}$ and applying the interpolation operator S^*, we get the following equation for $\boldsymbol{Q}$.

$$-S^*\nabla\Delta_0^{-1}S(\boldsymbol{Q}\cdot\boldsymbol{n}) - S^*\mathcal{L}^{-1}\mathbb{P}\Big(\eta S\boldsymbol{Q} + \mu\nabla\cdot(S\boldsymbol{A})\Big) + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b - S^*\mathcal{L}^{-1}\mathbb{P}\tilde{\boldsymbol{g}}, \tag{6.43}$$

where we recall that $\boldsymbol{A}$ is a function of $\boldsymbol{Q}$, given by $\boldsymbol{A} \equiv \boldsymbol{Q}\otimes\boldsymbol{n} + \boldsymbol{n}\otimes\boldsymbol{Q}$. As we did for the elliptic PDE, we solve this equation for $\boldsymbol{Q}$ using `gmres` and then solve Equations (6.39b) and (6.39b) for the velocity $\boldsymbol{u}$ and pressure p.

6.3.3   Solution to discrete system for Stokes equation

We now look at the IBDL formulation for the Stokes equation, given by

$$\mu\Delta\boldsymbol{u} - \nabla p + \eta S\boldsymbol{Q} + \mu\nabla\cdot(S\boldsymbol{A}) = \tilde{\boldsymbol{g}} \qquad \text{in } \mathcal{C} \tag{6.44a}$$

$$\nabla\cdot\boldsymbol{u} + S(\boldsymbol{Q}\cdot\boldsymbol{n}) = 0 \qquad\qquad \text{in } \mathcal{C} \tag{6.44b}$$

$$S^*\boldsymbol{u} + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b \qquad\qquad \text{on } \Gamma, \tag{6.44c}$$

where $\eta > 0$ for an exterior PDE domain Ω, and $\eta = 0$ for an interior Ω. In this case, we must again adjust the method of solution since the periodic Laplacian is not invertible. Additionally, note that as discussed for the Poisson equation in Section 5.3.3, for the case of $\eta = 0$, the constraint matrix has a one-dimensional nullspace, but since the eigenfunction spanning the nullspace again gives values of 0 on the physical domain, we need not alter the method for an interior domain. For an exterior domain, the addition of the single layer term results in a matrix that is invertible. Therefore, we need only to handle the nullspace of the periodic Laplacian.

We begin by decomposing the solution $\boldsymbol{u}$ as

$$\boldsymbol{u} = \boldsymbol{u}_0 + \bar{\boldsymbol{u}}, \tag{6.45}$$

where $\boldsymbol{u}_0$ has mean 0 on $\mathcal{C}$ and $\bar{\boldsymbol{u}} = (\bar{u}, \bar{v})$ gives the mean values of u and v, respectively. Following the same process as the previous section and noting that $\Delta\bar{\boldsymbol{u}} = 0$, we get that the equation that must be solved for $\boldsymbol{u}_0$ on $\mathcal{C}$ is given by

$$\mu\Delta\boldsymbol{u}_0 = -\mu\nabla\Delta_0^{-1}\Delta S(\boldsymbol{Q}\cdot\boldsymbol{n}) + \mathbb{P}\Big(\tilde{\boldsymbol{g}} - \eta S\boldsymbol{Q} - \mu\nabla\cdot(S\boldsymbol{A})\Big). \tag{6.46}$$

By integrating this over a general computational domain $\mathcal{C}$, we get

$$\mu\int_{\mathcal{C}}\Delta\boldsymbol{u}_0 d\boldsymbol{x} = -\mu\nabla\Delta_0^{-1}\Delta\int_{\mathcal{C}}S(\boldsymbol{Q}\cdot\boldsymbol{n})d\boldsymbol{x} - \nabla\Delta_0^{-1}\int_{\mathcal{C}}\nabla\cdot\Big(\tilde{\boldsymbol{g}} - \eta S\boldsymbol{Q} - \mu\nabla\cdot(S\boldsymbol{A})\Big)d\boldsymbol{x}$$
$$- \int_{\mathcal{C}}\Big(\eta S\boldsymbol{Q} + \mu\nabla\cdot(S\boldsymbol{A}) - \tilde{\boldsymbol{g}}\Big)d\boldsymbol{x}. \tag{6.47}$$

Using the divergence theorem where appropriate, we get

$$\mu\int_{\partial\mathcal{C}}\nabla\boldsymbol{u}_0\cdot\boldsymbol{n}_{\mathcal{C}}dl(\boldsymbol{x}) = -\mu\Delta_0^{-1}\Delta\int_{\partial\mathcal{C}}S(\boldsymbol{Q}\cdot\boldsymbol{n})\boldsymbol{n}_{\mathcal{C}}dl(\boldsymbol{x}) - \nabla\Delta_0^{-1}\int_{\partial\mathcal{C}}\tilde{\boldsymbol{g}}\cdot\boldsymbol{n}_{\mathcal{C}}dl(\boldsymbol{x})$$
$$+ \nabla\Delta_0^{-1}\int_{\partial\mathcal{C}}\Big(\eta S\boldsymbol{Q} + \mu S\boldsymbol{A}\Big)\cdot\boldsymbol{n}_{\mathcal{C}}dl(\boldsymbol{x}) - \eta\int_{\mathcal{C}}S\boldsymbol{Q}d\boldsymbol{x} - \mu\int_{\partial\mathcal{C}}(S\boldsymbol{A})\cdot\boldsymbol{n}_{\mathcal{C}}dl(\boldsymbol{x}) + \int_{\mathcal{C}}\tilde{\boldsymbol{g}}d\boldsymbol{x}$$
$$\tag{6.48}$$

If we assume Γ is away from the boundary of $\mathcal{C}$, the first, third, and fifth integrals on the right-hand-side vanish. If we take $\mathcal{C}$ to be a periodic box, the left-hand-side also vanishes, leaving us with

$$0 = -\nabla\Delta_0^{-1}\int_{\partial\mathcal{C}}\tilde{\boldsymbol{g}}\cdot\boldsymbol{n}_{\mathcal{C}}dl(\boldsymbol{x}) - \eta\int_{\mathcal{C}}S\boldsymbol{Q}d\boldsymbol{x} + \int_{\mathcal{C}}\tilde{\boldsymbol{g}}d\boldsymbol{x} \tag{6.49}$$

Interior PDE domain. Let us now first consider the case in which the PDE domain Ω is an interior domain. In this case, $\eta = 0$, and the second integral in Equation (6.49) vanishes. We are left with

$$\nabla\Delta_0^{-1}\int_{\partial\mathcal{C}}\tilde{\boldsymbol{g}}\cdot\boldsymbol{n}_{\mathcal{C}}dl(\boldsymbol{x}) = \int_{\mathcal{C}}\tilde{\boldsymbol{g}}d\boldsymbol{x}. \tag{6.50}$$

103

If we make the same choice for g_e as we did for the Poisson equation,

$$g_e = -\frac{1}{|\mathcal{C} \setminus \Omega|} \int_\Omega g d\boldsymbol{x}, \tag{6.51}$$

then the right-hand-side is 0, and since on $\partial\mathcal{C}$, $\tilde{\boldsymbol{g}}$ is a constant, the periodicity of $\mathcal{C}$ gives us 0 on the left-hand-side. With this solvability constraint satisfied, the solution to Equation (2.20) can be found, and it is unique up to an additive constant. We therefore find the solution with mean 0 on $\mathcal{C}$, and the potential strength, $\boldsymbol{Q}$, enforces the boundary condition, giving the correct solution on the PDE domain Ω. Then by inverting Δ in Equation (6.46) and applying the interpolation operator S^*, we get the following equation for $\boldsymbol{Q}$.

$$-S^*\Delta_0^{-1}\nabla\Delta_0^{-1}\Delta S(\boldsymbol{Q} \cdot \boldsymbol{n}) - S^*\Delta_0^{-1}\mathbb{P}\nabla \cdot (S\boldsymbol{A}) + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b - \frac{1}{\mu}S^*\Delta_0^{-1}\mathbb{P}\tilde{\boldsymbol{g}}. \tag{6.52}$$

After solving this equation using `gmres`, we then solve Equations (6.8a) and (6.8b) for p and $\boldsymbol{u}$.

Exterior PDE domain. Now let us consider the case in which Ω is an exterior domain. Returning to Equation (6.49), we then have that $\tilde{\boldsymbol{g}}$ on $\partial\mathcal{C}$ is $\boldsymbol{g}$, the periodic forcing function in the PDE. Therefore, the first integral vanishes. Using that $\int_\mathcal{C} S\boldsymbol{Q} d\boldsymbol{x} = \int_\Gamma \boldsymbol{Q} ds$, we then get the solvability constraint

$$\eta \int_\Gamma \boldsymbol{Q} ds = \int_\mathcal{C} \tilde{\boldsymbol{g}} d\boldsymbol{x}. \tag{6.53}$$

We therefore use this constraint to solve for the unknown $\bar{\boldsymbol{u}}$, as we did for the IBSL method in Sections 2.2.4 and 2.3.3. Discretizing this constraint, we get the following system of equations

$$\mu\Delta\boldsymbol{u}_0 - \nabla p + \eta S\boldsymbol{Q} + \mu\nabla \cdot (S\boldsymbol{A}) = \tilde{\boldsymbol{g}} \tag{6.54a}$$

$$\nabla \cdot \boldsymbol{u}_0 + S(\boldsymbol{Q} \cdot \boldsymbol{n}) = 0 \tag{6.54b}$$

$$S^*\boldsymbol{u}_0 + \bar{\boldsymbol{u}}\mathbb{1}_{N_{IB}} + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b \tag{6.54c}$$

$$\eta(\Delta s)\mathbb{1}_{N_{IB}}^\mathsf{T}\boldsymbol{Q} = (\Delta x\Delta y)\mathbb{1}_{N^2}^\mathsf{T}\tilde{\boldsymbol{g}} \tag{6.54d}$$

We therefore have the following discrete system of equations,

$$- S^* \Delta_0^{-1} \nabla \Delta_0^{-1} \Delta S (\boldsymbol{Q} \cdot \boldsymbol{n}) - S^* \Delta_0^{-1} \mathbb{P} \Big(\frac{\eta}{\mu} S \boldsymbol{Q} + \nabla \cdot (S \boldsymbol{A}) \Big) + \bar{\boldsymbol{u}} \mathbb{1}_{N_{IB}} + \frac{1}{2} \boldsymbol{Q}$$

$$= \boldsymbol{U}_b - \frac{1}{\mu} S * \Delta_0^{-1} \mathbb{P} \tilde{\boldsymbol{g}} \quad (6.55\text{a})$$

$$\eta (\Delta s) \mathbb{1}_{N_{IB}}^\mathsf{T} \boldsymbol{Q} = (\Delta x \Delta y) \mathbb{1}_{N^2}^\mathsf{T} \tilde{\boldsymbol{g}}, \tag{6.55b}$$

where the unknowns are $\boldsymbol{Q}$ and $\bar{\boldsymbol{u}}$. We solve this equation using `gmres`, and then solve Equations (6.39a) and (6.39b) for p and $\boldsymbol{u}_0$. Finally, we find $\boldsymbol{u}$ by adding $\bar{\boldsymbol{u}}$.

6.4 Results: Stokes and Brinkman equations

In this section, we apply the Immersed Boundary Double Layer method to Stokes and Brinkman equations with Dirichlet boundary conditions and discuss numerical results. In Section 6.4.1, we revisit the Brinkman equation seen in Section 2.4.2 and demonstrate the improved efficiency of the IBDL method. In Section 6.4.2, we examine the affect of different choices of η in the completed IBDL formulation. In Section 6.4.3, we observe the affects of using two finite difference forms for the Laplacian. In Section 6.4.4, we apply the method to the problem of Stokes flow past a periodic array of cylinders and compare to previous numerical and asymptotic results. Lastly, in Section 6.4.5, we apply the IBDL and IBSL methods to flow past 9 objects of varying size to demonstrate that as the domain increases in complexity, the efficiency of the IBDL method over the IBSL method can be even more extreme.

6.4.1 Brinkman equation with analytical solution

We first revisit the Brinkman PDE from Section 2.4.2, given by

$$\Delta \boldsymbol{u} - \boldsymbol{u} - \nabla p = \boldsymbol{g} \qquad \text{in } \Omega \tag{6.56a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{6.56b}$$

$$\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma, \tag{6.56c}$$

where we use the analytical solution given by

$$u = e^{\sin x} \cos y \tag{6.57a}$$

$$v = -\cos x \, e^{\sin x} \sin y \tag{6.57b}$$

$$p = e^{\cos y} \tag{6.57c}$$

to find the boundary values, $\boldsymbol{U}_b = \boldsymbol{u}|_\Gamma$, and the forcing function,

$$\boldsymbol{g} = \begin{pmatrix} e^{\sin x} \cos y \, (\cos^2 x - \sin x - 2) \\ -\cos x \, e^{\sin x} \sin y \, (\cos^2 x - 3\sin x - 3) + \sin y \, e^{\cos y} \end{pmatrix}. \tag{6.58}$$

Here Ω is the interior of a circle of radius 0.75, centered at the origin, and the computational domain is the periodic box, $\mathcal{C} = [-1, 1]^2$. We use a Fourier spectral method and equally spaced boundary points with $\Delta s \approx \alpha \Delta x$ for various values of α, and the solutions are computed for grid sizes ranging from $N = 2^6$ to 2^{12}. For the IBDL method, we use interpolation for grid points within $m_1 = 2(\log_2 N - 4)$ meshwidths of Γ.

Table 6.1 gives the iteration counts for using `minres` and `gmres` to solve Equation (6.56) with the IBSL and IBDL methods, respectively. As expected from the formulation of the IBDL method, we see that the IBDL iteration counts are small and essentially constant as we refine the mesh or boundary point spacing. Furthermore, while the IBSL method fails to reach the desired tolerance for the $\Delta s \approx 0.75 \Delta x$, we have no such problem with the IBDL method. Figures 6.1a-6.1b demonstrate the first-order convergence of the velocity solutions, and again, the error sizes are comparable

Iteration Counts - Equation (6.56)							
	$\Delta s \approx 2\Delta x$		$\Delta s \approx 1.5\Delta x$		$\Delta s \approx 1\Delta x$		$\Delta s \approx 0.75\Delta x$	
Δx	IBSL	IBDL	IBSL	IBDL	IBSL	IBDL	IBSL	IBDL
2^{-5}	123	10	296	10	2558	10	16190 stag.	10
2^{-6}	185	10	472	10	5578	10	47863 stag.	10
2^{-7}	257	10	678	10	14438	10	95009 stag.	9
2^{-8}	351	10	825	10	17475	10	68861 stag.	9
2^{-9}	478	10	1137	10	17789	10	62697 stag.	9
2^{-10}	622	10	1620	10	20333	10	75910 stag.	9
2^{-11}	821	10	2180	10	27467	10	79699 stag.	9

TABLE 6.1. Number of iterations of `minres` and `gmres`, with tolerance 10^{-8}, needed to solve Equation (6.56) using the IBSL and IBDL methods, respectively, on the periodic computational domain $[-1, 1]^2$. The word *stag.* is used to indicate that the method stagnated at this number of iterations without converging to the desired tolerance.

between the two methods. Through this example, we once again see the greater efficiency of the IBDL method.

We also include refinement studies for the pressure for the IBSL and IBDL methods. Figure 6.1c shows the refinement studies for both methods on the entire PDE domain Ω. Since the pressure is discontinuous with the IBSL method, it fails to converge pointwise and gets lower order convergence for the L^2 norm. However, in the IBDL method, the pressure actually blows up near the boundary, giving no convergence. Figure 6.1d, on the other hand, uses the errors for the portion of Ω that is away from Γ by a distance of 0.02. We can see that both solutions give first-order convergence of the pressure away from Γ by this distance.

6.4.2 Choice of η

For this section, we use the following exterior Brinkman equation to explore the choice of η in the completed IBDL method. The PDE is

$$\Delta \boldsymbol{u} - k^2 \boldsymbol{u} - \nabla p = \boldsymbol{g} \qquad \text{in } \Omega \tag{6.59a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{6.59b}$$

$$\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma, \tag{6.59c}$$

(a) IBDL, u

(b) IBDL, v

(c) pressure, Ω
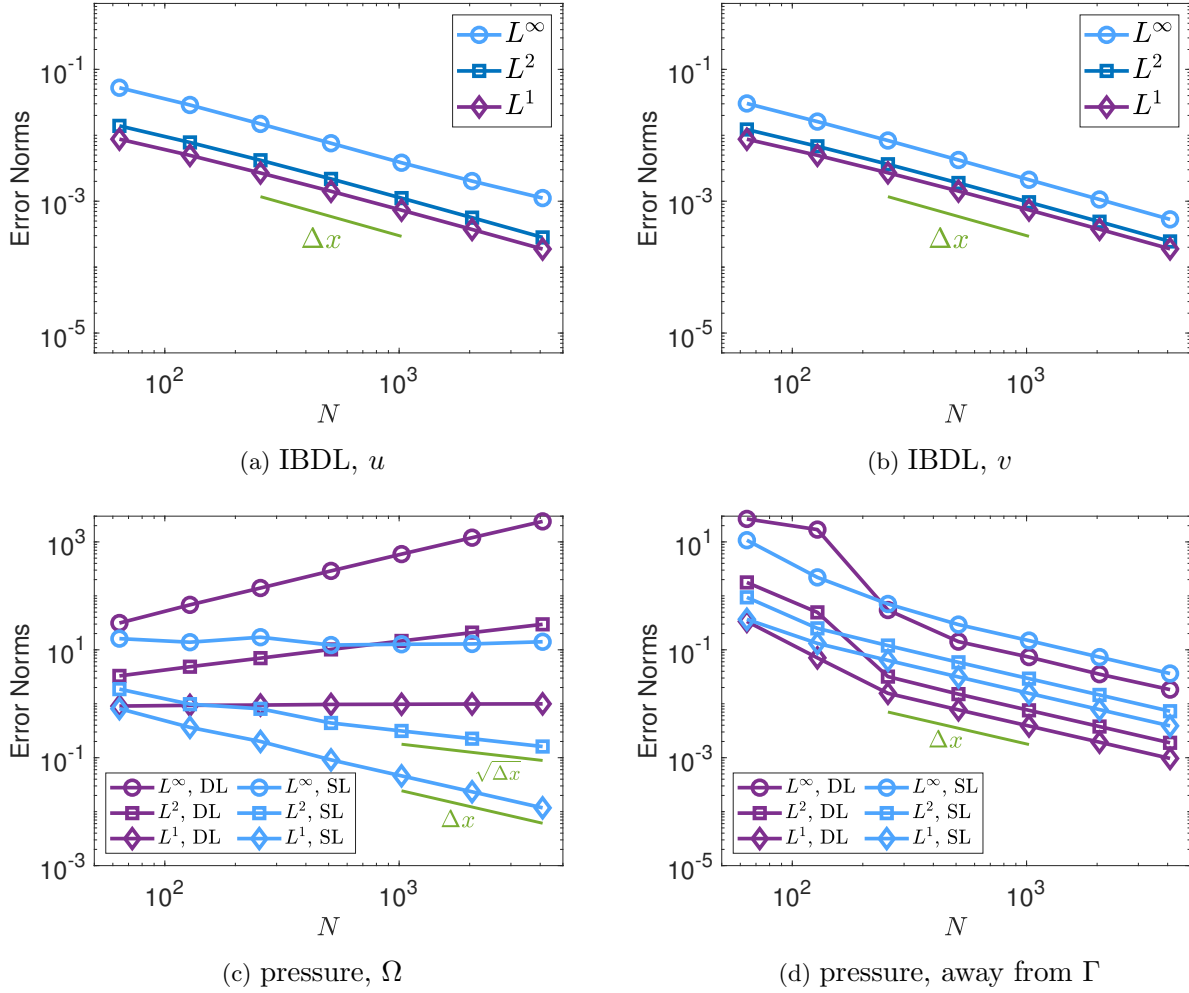
(d) pressure, away from Γ

FIGURE 6.1. Refinement studies for solutions to Equation (6.56) found using the IBSL and IBDL methods. The computational domain is the periodic box $[-1, 1]^2$, and the boundary point spacing is $\Delta s \approx 1.5\Delta x$. We use a Fourier spectral method, and for IBDL, we use interpolation for grid points within $m_1 = 2(\log_2 N - 4)$. Figures 6.1a-6.1b give the IBDL refinement studies for velocity. Figure 6.1c gives the refinement study for pressure on Ω for both methods, and Figure 6.1d gives the refinement study for pressure on the portion of Ω that is away from Γ by a distance of 0.02. All errors are absolute.

where we use the analytical solution given by

$$u = e^{\sin(\pi x)} \cos(\pi y) \tag{6.60a}$$

$$v = -\cos(\pi x)\, e^{\sin(\pi x)} \sin(\pi y) \tag{6.60b}$$

$$p = e^{\cos(\pi y)} \tag{6.60c}$$

108

to determine the boundary values, $\boldsymbol{U}_b = \boldsymbol{u}|_\Gamma$, and the forcing function,

$$\boldsymbol{g} = \begin{pmatrix} \pi^2 e^{\sin(\pi x)} \cos(\pi y) \left(\cos^2(\pi x) - \sin(\pi x) - 1 - k^2/\pi^2 \right) \\ -\pi^2 \cos(\pi x) e^{\sin(\pi x)} \sin(\pi y) \left(\cos^2(\pi x) - 3\sin(\pi x) - 2 - k^2/\pi^2 \right) + \pi \sin(\pi y) e^{\cos(\pi y)} \end{pmatrix}.$$
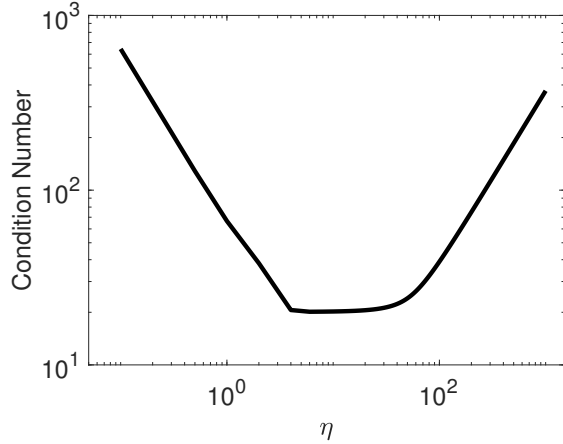
(6.61)

Here, we will use $k = 0, 0.1$, and 3. Ω is the region of the periodic box, $\mathcal{C} = [-1, 1]^2$ that is exterior to a circle of radius 0.75, centered at the origin. We use a Fourier spectral method with equally spaced boundary points with $\Delta s \approx \Delta x$, and we use interpolation for grid points within $m_1 = 2(\log_2 N - 4)$ of Γ.

We begin with Stokes equation, or $k = 0$. In this case, in order to represent flows with net force and net torque, we must use the completed IBDL method, with $\eta > 0$. Figure 6.2a shows the L^∞ refinement studies for this problem for four different values of η. We can see that using a larger value of η can result in lower pointwise error on fine grids. For this problem, there is no meaningful benefit to increasing η above 10. We have observed this to be true for other problems as well. For a range of η values, Figure 6.2b plots the condition number of the operator that must be inverted to solve for $\boldsymbol{Q}$. This operator is seen in Equation (6.55). Selecting a very small η value will result in a large condition number, as the operator approaches that for the IBDL without the completion. The smallest condition number occurs at $\eta \approx 6$. After this, the condition number grows as we increase η. However, for moderate values of η there is not a large impact on the iteration counts. For this problem, the iteration count for $\eta = 10$ is about 8, and the iteration count for $\eta = 100$ is still only about 13. To compare, the condition number for the IBSL operator for $N = 2^7$ and $\Delta s \approx \Delta x$ is approximately $2.6(10)^9$. For the wider boundary point pacing of $\Delta s \approx 2\Delta x$, the condition number is approximately $1.7(10)^7$. These condition numbers are much higher than any of the condition numbers we see for η values ranging from 1 to 1000. Additionally, finer grids would result in even larger condition numbers for the IBSL method. Therefore, we see that there is flexibility in the choice of η, and we need not fear approaching the poor conditioning of the IBSL method. For the majority of this dissertation, we use $\eta = 10$.
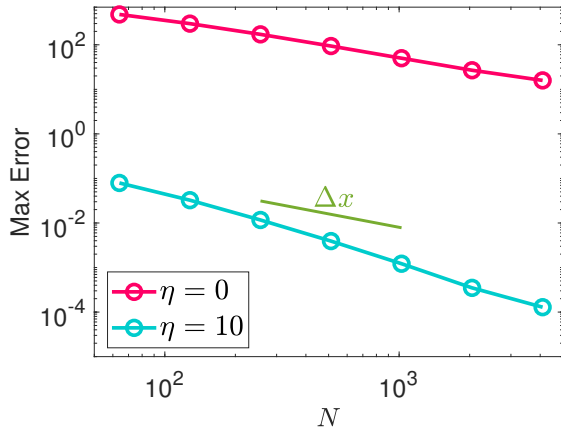
We next look at the Brinkman equation for $k = 0.01$ and $k = 3$. Figures 6.2c-6.2d show the refinement studies for utilizing the IBDL method without the completion and with the completion
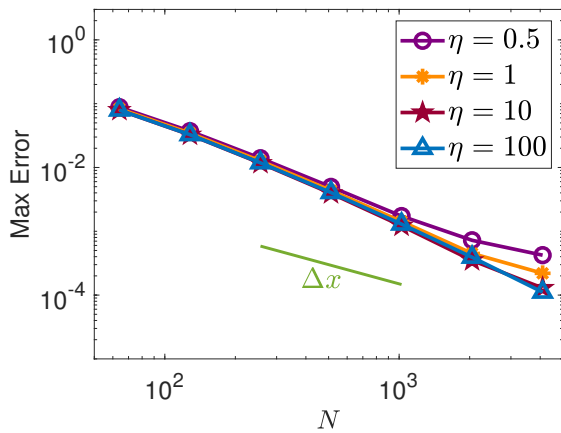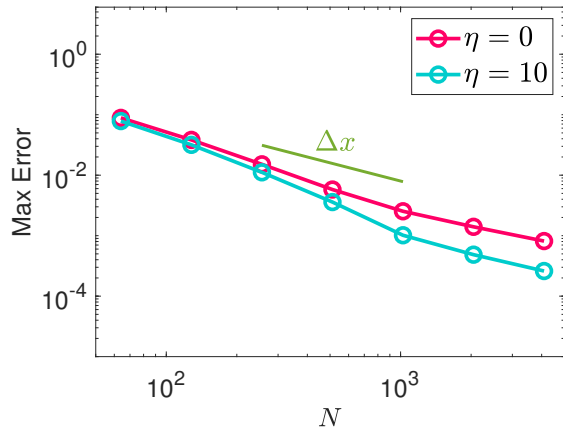
(a) Stokes equation

(b) Stokes equation

(c) Brinkman, $k = 0.01$

(d) Brinkman, $k = 3$

FIGURE 6.2. Figures 6.2a, 6.2c, and 6.2d give L^∞ refinement studies for horizontal velocity solutions to the exterior Stokes and Brinkman equations (6.59) using the completed IBDL method and various values of η. Figure 6.2b gives a plot of the condition number versus η for operator that must be inverted for Stokes equation. Ω is the region of periodic box, $\mathcal{C} = [-1, 1]^2$ that is exterior to a circle of radius 0.75, centered at the origin. The method uses a Fourier spectral method with equally spaced boundary points with $\Delta s \approx \Delta x$, and interpolation is used for grid points within $m_1 = 2(\log_2 N - 4)$ of Γ. All errors are absolute.

for $\eta = 10$. We can see that for a small k value, we may still obtain convergence without the completion, but including the single layer term is important to obtain small errors. Using $\eta = 10$ in this case drastically lowers the errors. For larger k values, as the Brinkman equation diverges from Stokes equation, the need to include the single layer term diminishes, but including it often lowers the error, as we see for $k = 3$.

6.4.3 Finite difference method

For this section, we use the following PDE to examine the results from the IBDL method with two different finite difference discretizations for $\nabla \cdot \nabla$. The PDE is

$$\Delta \boldsymbol{u} - \nabla p = \boldsymbol{g} \qquad \text{in } \Omega \tag{6.62a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{6.62b}$$

$$\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma, \tag{6.62c}$$

where we use the analytical solution given by

$$u = \sin y - xe^{xy} \tag{6.63a}$$

$$v = \cos x + ye^{xy} \tag{6.63b}$$

$$p = e^{x+y} \tag{6.63c}$$

to determine the boundary values, $\boldsymbol{U}_b = \boldsymbol{u}|_\Gamma$, and the forcing function,

$$\boldsymbol{g} = \begin{pmatrix} -e^{xy}\left(2y + xy^2 + x^3\right) - \sin y - e^{x+y} \\ e^{xy}\left(2x + x^2 y + y^3\right) - \cos x - e^{x+y} \end{pmatrix}. \tag{6.64}$$

Here Ω is the interior of a circle of radius 0.25, centered at the origin. The computational domain is $\mathcal{C} = [-0.5, 0.5]^2$. We use equally spaced boundary points with $\Delta s \approx 0.75\Delta x$, and we use interpolation for grid points within $m_1 = 6$ meshwidths of Γ. We also use the smoother 6-point B-spline delta function defined in Section 6.3.1.

As discussed in Section 2.3.1, the discrete divergence of the discrete gradient results in a Laplacian with a wider stencil. In Method 1, we invert this wide Laplacian when solving for pressure, as we do in the IBSL method. Figure 6.3a shows the refinement study for horizontal velocity, and Figure 6.3c shows the horizontal velocity error for the grid size $N = 2^{12}$. Figure 6.3e shows the pressure solution for grid size $N = 2^7$. To plot the pressure, we again find the difference between the computed pressure and analytical solution at the origin and add this constant to the computed pressure. We do not achieve first-order convergence, and Figure 6.3c shows larger pointwise errors,
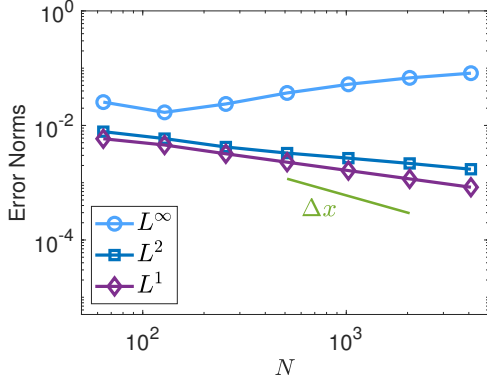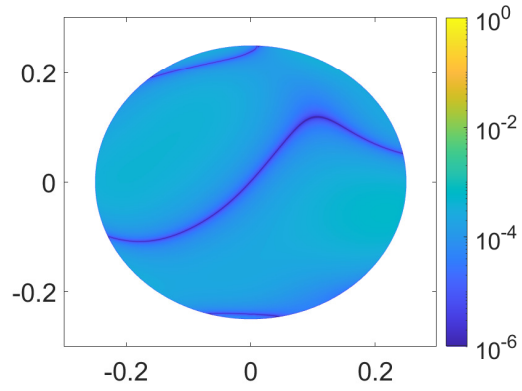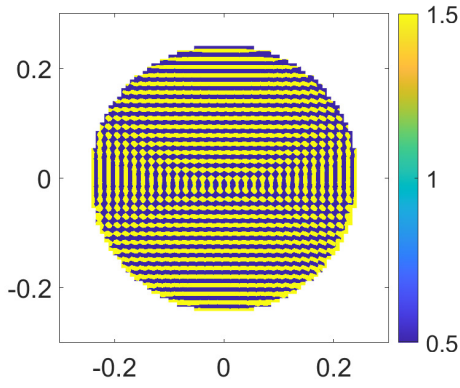
(a) refinement, method 1
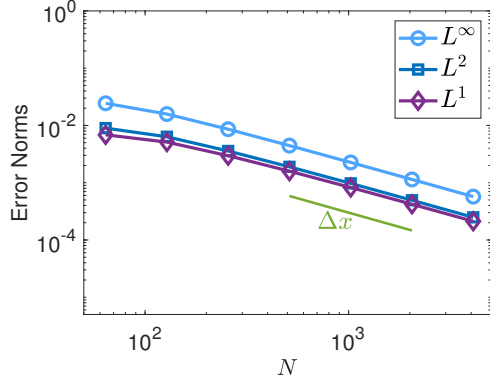
(b) refinement, method 2
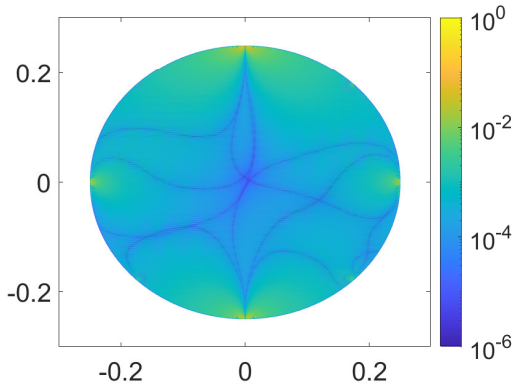
(c) error, method 1

(d) error, method 2

(e) pressure, method 1
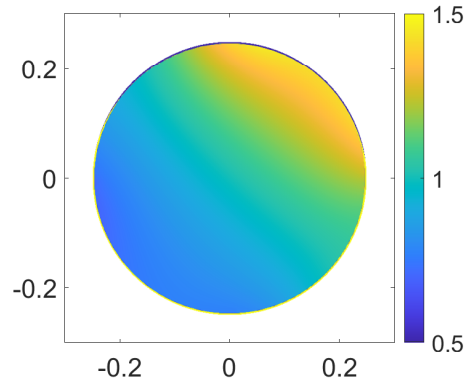
(f) pressure, method 2

FIGURE 6.3. Refinement studies and plots for Equation (6.62), found using the IBDL method with the two forms of finite difference discretizations described in Section 6.4.3. Figures 6.3a-6.3b give the refinement studies for horizontal velocity, and Figures 6.3c-?? show the horizontal velocity errors for $N = 2^{13}$. Figures 6.3e-6.3f give the pressure solutions for $N = 2^7$. Ω is the interior of a circle of radius 0.25, centered at the origin, and the computational domain is $\mathcal{C} = [-0.5, 0.5]^2$. The methods use the 6-point B-spline delta function defined in Section 6.3.1 and equally spaced boundary points with $\Delta s \approx 0.75\Delta x$. Interpolation is used for grid points within $m_1 = 6$ meshwidths of Γ. The errors used in the refinement studies are absolute, and Figures 6.3c-6.3f use a grid size of $N = 2^{12}$.

especially as we near the boundary, and we note that increasing the interpolation width is not sufficient to recover pointwise convergence. Furthermore, Figure 6.3e demonstrates that the interaction between the two discrete Laplacian operators case high frequency oscillations in the pressure solution.

In Method 2, we approximate the discrete divergence of the discrete gradient with the usual 5-point, second-order accurate discrete Laplacian. Figures 6.3b, 6.3d, and 6.3f give the corresponding refinement study, velocity error, and pressure solution. We can see that this approximation, combined with the smoother delta function, gives us first-order convergence of the solution and avoids the high frequency oscillations in the pressure. More investigation into other possible finite difference discretizations is needed, but Method 2 provides a simple option.

6.4.4 Stokes flow around a periodic array of cylinders

We next look at steady Stokes flow around a doubly periodic array of cylinders. This problem has been studied at length, and we can compare our numerical results for the drag force on the cylinder to literature and asymptotic approximations [**23**, **32**, **35**, **52**]. To do this, we look at the PDE

$$\Delta \boldsymbol{u} - \nabla p = \boldsymbol{g} \qquad \text{in } \Omega \tag{6.65a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{6.65b}$$

$$\boldsymbol{u} = 0 \qquad \text{on } \Gamma, \tag{6.65c}$$

where Ω is the region in the periodic box $[-0.5, 0.5]^2$ that is exterior to a disk centered at the origin, and Γ is the boundary of this disk. Let us define c as the fraction of the two-dimensional area occupied by the disk. Here, $c = \pi r^2$, and the maximum value is $c_{max} = \pi/4 \approx 0.785$. We explore the dimensionless drag force per unit length of a cylinder obtained from the IBDL method for different values of c. Let the fluid viscosity be μ, which is chosen as 1 in Equation (6.65a). Let U be the average velocity in the x-direction on the left side of the periodic box. In other words,

$$U = \int_{-0.5}^{0.5} u(-0.5, y) dy. \tag{6.66}$$

We drive the flow with the forcing function

$$\boldsymbol{g} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \tag{6.67}$$

and numerically estimate the integral for U. Recall that in Chapter 3, we denoted the net force on an object as $\boldsymbol{B}$. For the IBDL method, we solve Equation (6.65) using the completed formulation described in Section 6.1.3. We therefore add a supplementary flow provided by a single layer potential with strength $\eta \boldsymbol{Q}$, with $\eta = 10$. Then, as discussed in Section 6.1.3, we calculate the net force by numerically approximating

$$\boldsymbol{B} = -\eta \int_{\Gamma} \boldsymbol{Q}(s) ds. \tag{6.68}$$

Similarly, as discussed in Section 3.5.2, we calculate the net force in the IBSL case by numerically approximating

$$\boldsymbol{B} = - \int_{\Gamma} \boldsymbol{F}(s) ds. \tag{6.69}$$

Then we calculate the dimensionless drag force, $D = B_1/(\mu U)$. We compare the values of D from the IBDL method with those obtained from the IBSL method and the integral method of Greengard and Kropinski [23]. We also compare the results to the asymptotic expressions for the dilute ($c \ll 1$) and dense ($c_{max} - c \ll 1$) cases [32, 52]. These expressions are given by

$$D_{den} \approx \frac{9\pi}{2\sqrt{2}} \left(1 - \sqrt{\frac{c}{c_{max}}} \right)^{-5/2} \tag{6.70}$$

$$D_{dil} \approx \frac{8\pi}{\log 1/c - 1.47633597 + 2c - 1.77428264c^2 + 4.07770444c^3 - 4.84227402c^4}. \tag{6.71}$$

Table 6.2 gives the dimensionless drag forces for the three methods, and Figure 6.4 plots D versus c for the IBDL method and the asymptotic approximations. For the IBDL and IBSL methods, we use a Fourier spectral method with a boundary point spacing of $\Delta s \approx \Delta x$ and a mesh size of $N = 2^{13}$. We see very good agreement with the literature and asymptotic results, especially for the dilute case. For $c \leq 0.4$, the relative errors for the IBDL method compared

114

Dimensionless Drag, D					
c	IBDL	IBSL	G&K	D_{dil}	D_{den}
0.05	15.5591	15.5648	15.5578	15.5578	–
0.1	24.8368	24.8431	24.8317	24.8323	–
0.2	51.5459	51.5553	51.5269	51.6068	–
0.3	102.940	102.953	102.881	105.234	–
0.4	218.094	218.093	217.894	270.011	–
0.5	533.420	533.226	532.548	–	544.313
0.6	1770.23	1767.14	1763.57	–	1775.23
0.7	13775.8	13580.9	13519.3	–	13512.3
0.71	19087.8	18706.3	–	–	18607.8
0.72	27734.8	26961.8	–	–	26785.0
0.75	147482	127572	127543	–	127424

TABLE 6.2. Dimensionless drag, D, for various values of area concentration, c, for Stokes flow past a periodic array of cylinders. Included are results from the IBDL method, the IBSL method, and the boundary integral method of Greengard and Kropinski [23]. The last two columns give dilute and dense asymptotic approximations [32, 52].
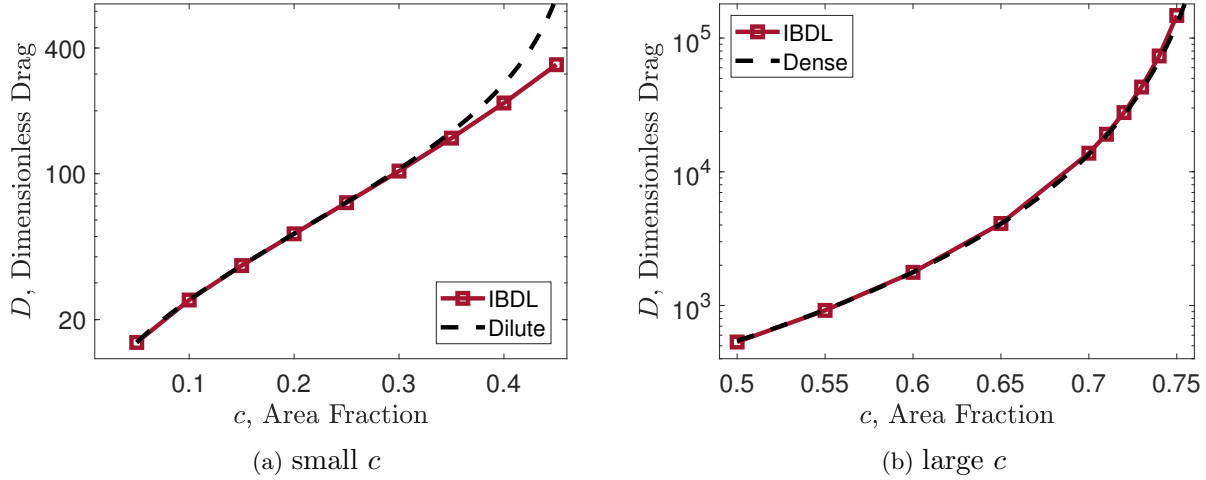

FIGURE 6.4. Dimensionless drag, D for different area concentrations for Stokes flow past a periodic array of cylinders. Included are results from the IBDL method and the dilute and dense asymptotic approximations. For the IBDL method, we use a grid size of $N = 2^{13}$, boundary point spacing of $\Delta s \approx \Delta x$, and interpolation width $m_1 = 2(\log_2 N - 4)$.

to the Greengard and Kropinski values are less than 0.1%. For $c \leq 0.6$, the relative error is less than 1%, and for $c = 0.7$, less than 2%. The IBSL method does outperform our method for a very dense periodic array of cylinders, but we are able to obtain reasonable results for up to $c \approx 0.72$, with a relative error to the asymptotic approximation under 5%. The relative gap between cylinders is then approximately 0.04. Kallemov et al. report a bound on the relative gap
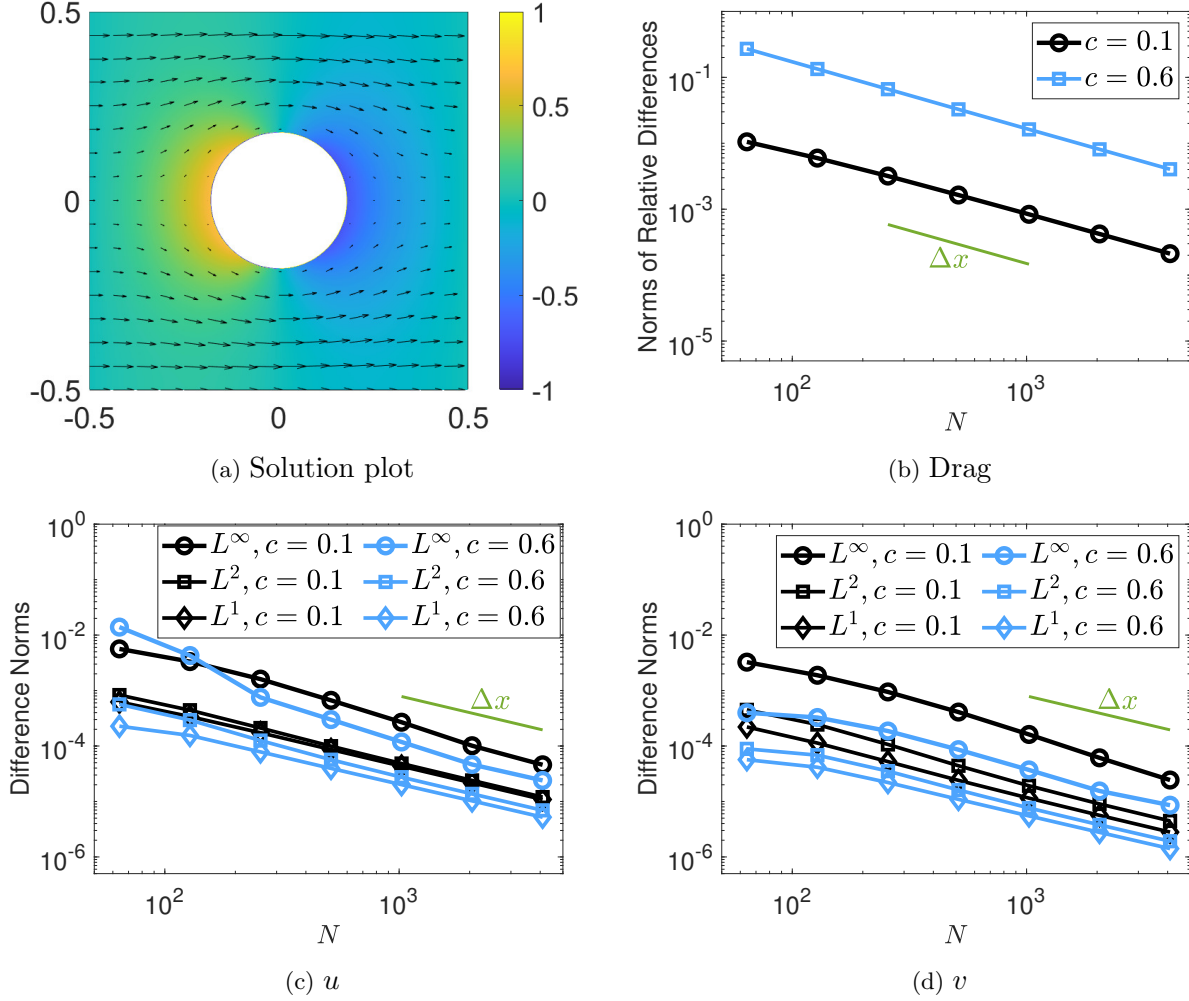
(a) Solution plot

(b) Drag

(c) u

(d) v

FIGURE 6.5. Solution plot and solution and drag refinement studies for flow past a periodic array of cylinders. A Fourier spectral method is used. The computational domain is the periodic box $[-0.5, 0.5]^2$, the boundary point spacing is $\Delta s \approx \Delta x$, the interpolation width is $m_1 = 2(\log_2 N - 4)$, and $\eta = 10$. Figure 6.5a shows the solution for grid size 2^{10} and $c = 0.1$. Figure 6.5b shows the refinement study for the dimensionless drag, D, using the difference of successive values, relative to the value of the drag. Figures 6.5c-6.5d show refinement studies for the velocity components, using the absolute differences of successive solutions. The refinement studies in Figures 6.5b-6.5d are for $c = 0.1$ and $c = 0.6$.

for their preconditioned IBSL-like method of approximately 0.02 [**35**]. Again, it is worth noting that the method of Kallemov et al. and the IBSL method are able to achieve a much more accurate drag in the case of $c = 0.75$, or a relative gap of 0.02. The poor performance of the IBDL method when the cylinders are this close is worth further investigation, as there are applications in which we may wish to simulate objects with this close interaction.

Iteration Counts - 9 Ellipses					
	$\Delta s \approx 2\Delta x$		$\Delta s \approx 1.5\Delta x$		$\Delta s \approx \Delta x$	
Δx	IBSL	IBDL	IBSL	IBDL	IBSL	IBDL
2^{-4}	420	25	1058	25	4159	25
2^{-5}	509	25	1540	25	5906	25
2^{-6}	599	25	1490	26	9052	26
2^{-7}	641	26	1605	26	16251	26
2^{-8}	650	26	1957	26	24436	26
2^{-9}	845	26	2065	27	26790	27
2^{-10}	1117	27	2724	27	34571	27

TABLE 6.3. Number of iterations of `minres` and `gmres`, with tolerance 10^{-8}, needed to solve Equation (6.65) using the IBSL and IBDL methods, respectively, for Ω, the region exterior to 9 randomly placed ellipses, and $\mathcal{C} = [-2, 2]^2$. Both methods use a Fourier spectral method, and the IBDL method uses $\eta = 10$.

Figure 6.5a shows a plot of the IBDL solution for the case of $c = 0.1$. The color illustrates the pressure that has mean 0 on $\mathcal{C}$. This solution was obtained on a grid size of $N = 2^{10}$, and the velocity vectors are plotted every 2^6 grid points. Figures 6.5b-6.5d then show refinement studies for the dimensionless drag and velocity solutions for two representative flows, with $c = 0.1$ and $c = 0.6$. The refinement studies are done using the difference of successive solutions, and the drag refinement study uses these differences, relative to the drag values. We see first order convergence in all refinement studies. As usual, the number of iterations of `gmres` needed for these solutions is essentially independent of grid size. For $0.05 \leq c \leq 0.55$, about $9 - 15$ iterations were needed. For $c = 0.6$, about $16 - 18$ iterations were needed. For $c = 0.7$, about 28 iterations were needed, and for $c = 0.75$, about 44 iterations were needed.

6.4.5 Stokes flow around many objects

We next examine Stokes flow past a periodic array of 9 arbitrarily placed ellipses that differ in size. We again solve Equation (6.65), with Ω the region in the computational domain, $\mathcal{C} = [-2, 2]^2$, that is exterior to the 9 ellipses. For the IBDL method, we use an interpolation width of $m_1 = \log_2 N - 4$ and $\eta = 10$. We solve the equation using both the IBDL and IBSL methods, with a Fourier spectral discretization, using a boundary point spacing of $\Delta s \approx \alpha \Delta x$ for various α values. Table 6.3 gives the iteration counts of the Krylov methods needed to solve with these methods. The wide boundary

point spacing of $\alpha = 2$ is generally used for the IBSL method in order to control the conditioning. However, we see that for this complex domain, the IBSL method requires high iteration counts, even for this wide spacing. When solving this PDE for the region exterior to one circle of radius 0.15, the IBSL method requires 58 iterations for $\alpha = 2$ and $\Delta x = 2^{-10}$. Moving to this more complicated domain, the number of iterations required is multiplied by more than 19. On the other hand, the iteration count for the IBDL method gets multiplied by less than 3. Therefore, we can see that the iteration counts resulting from the better conditioning of the IBDL method can become even more favorable when the domain is more complicated, such as when it contains more obstacles. We can see in Table 6.3 that the IBSL iteration counts are of course even more extreme for tighter boundary point spacing. In the case of moving objects or Navier-Stokes, where this solve would be required at each step in time, IBSL would be impractical without preconditioning. Figure 6.6 shows the solution plot for $N = 2^{10}$ and the IBDL velocity refinement studies. The color on the solution plot illustrates the pressure solution that has mean 0 on $\mathcal{C}$.

6.5   Results: Navier-Stokes equation

In this section, we implement the IBDL method for the Navier-Stokes equation, given by

$$\rho\Big(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\Big) = \mu \Delta \boldsymbol{u} - \nabla p \qquad \text{in } \Omega \tag{6.72a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{6.72b}$$

$$\boldsymbol{u} = \boldsymbol{U}_b \qquad \text{on } \Gamma. \tag{6.72c}$$

We utilize the completed IBDL method described in Section 6.1.3 and test the performance of the IBDL method by solving for a flow around a cylinder for Reynolds numbers 10 and 100. For $Re = 10$, we and compare the drag force against the IBSL method, and for $Re = 100$, we compare Strouhal number against the IB method presented by Lai and Peskin [38].

118
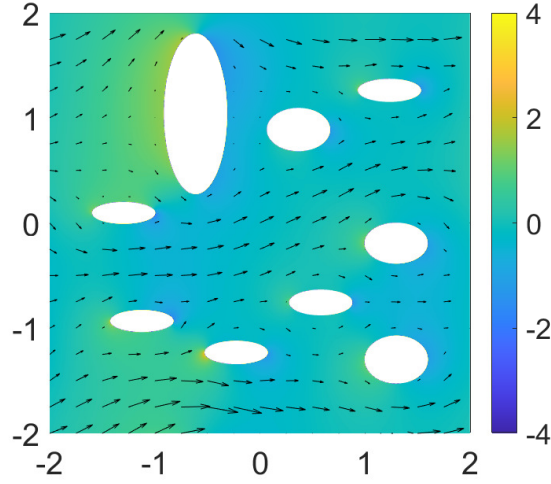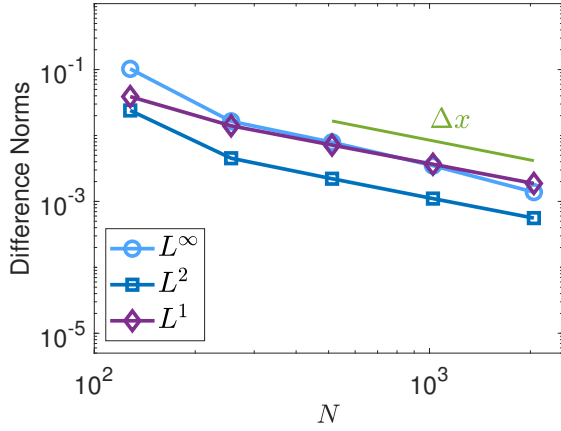
(a) Solution plot

(b) u

(c) v

FIGURE 6.6. Solution plot and refinement studies for Stokes flow past a periodic array of 9 randomly placed ellipses. The computational domain is the periodic box $[-2, 2]^2$, the boundary point spacing is $\Delta s \approx \Delta x$, the interpolation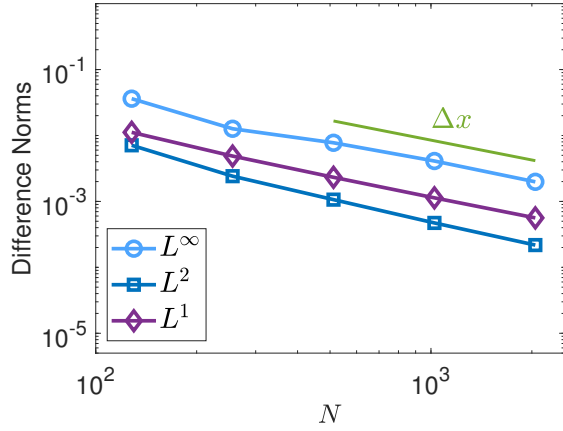 width is $m_1 = \log_2 N - 4$, $\eta = 10$, and a Fourier spectral method is used. Figure 6.6a shows the solution for grid size 2^{10}, and Figures 6.5c-6.5d show refinement studies for the velocity components, using the absolute difference of successive solutions.

6.5.1 Numerical implementation

For time discretization, we use an implicit-explicit scheme analyzed in [2] and used in an immersed boundary framework in [22]. In this scheme, the nonlinear terms are treated explicitly in time and the viscous and boundary distributions are treated implicitly in time. The PDE on $\mathcal{C}$ resulting

119

from the completed IBDL formulation is

$$\rho\Big(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\Big) = \mu \Delta \boldsymbol{u} - \nabla p + \eta S \boldsymbol{Q} + \mu \nabla \cdot (S\boldsymbol{A}) \qquad \text{in } \Omega \tag{6.73a}$$

$$\nabla \cdot \boldsymbol{u} + S(\boldsymbol{Q} \cdot \boldsymbol{n}) = 0 \qquad \text{in } \Omega \tag{6.73b}$$

$$S^* \boldsymbol{u} + \frac{1}{2}\boldsymbol{Q} = \boldsymbol{U}_b \qquad \text{on } \Gamma. \tag{6.73c}$$

Using the IMEX scheme, the discretized system is

$$\rho\Big(\frac{3\boldsymbol{u}^{n+1} - 4\boldsymbol{u}^n + \boldsymbol{u}^{n-1}}{2\Delta t} + 2\boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n - \boldsymbol{u}^{n-1} \cdot \nabla \boldsymbol{u}^{n-1}\Big)$$
$$= -\nabla p^{n+1} + \mu \Delta \boldsymbol{u}^{n+1} + \eta S \boldsymbol{Q}^{n+1} + \mu \nabla \cdot (S\boldsymbol{A}^{n+1}) \tag{6.74a}$$

$$\nabla \cdot \boldsymbol{u}^{n+1} + S(\boldsymbol{Q}^{n+1} \cdot \boldsymbol{n}^{n+1}) = 0 \tag{6.74b}$$

$$S^* \boldsymbol{u}^{n+1} + \frac{1}{2}\boldsymbol{Q}^{n+1} = \boldsymbol{U}_b^{n+1}. \tag{6.74c}$$

The first equation can be rewritten as

$$\Big(\mu \Delta - \frac{3\rho}{2\Delta t}\mathbb{I}\Big)\boldsymbol{u}^{n+1} - \nabla p^{n+1} + \eta S \boldsymbol{Q}^{n+1} + \mu \nabla \cdot (S\boldsymbol{A}^{n+1})$$
$$= \rho\Big(\frac{-4\boldsymbol{u}^n + \boldsymbol{u}^{n-1}}{2\Delta t} + 2\boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n - \boldsymbol{u}^{n-1} \cdot \nabla \boldsymbol{u}^{n-1}\Big), \tag{6.75}$$

where the right-hand side is known when advancing the system to the next time step. Therefore, by treating the nonlinearity explicitly in time, we recover the Brinkman equation, and we can use the completed IBDL method.

For the application to flow past a cylinder, we use the parameters from [38]. The computational domain is the periodic box $\mathcal{C} = [0, 8]^2$, and the cylinder of radius 0.15 is located at $(1.85, 4.0)$. To drive the flow, at each step in time, we reset the horizontal velocity to be $u_\infty = 1$ and the vertical velocity to be 0 for a strip along the left side of $\mathcal{C}$ that is 4 meshwidths wide. We define the Reynolds number as

$$Re = \frac{2r\rho u_\infty}{\mu}, \tag{6.76}$$

where r is the radius of the cylinder. We use a time step of $\Delta t = 1.8(10)^{-3}$, and a mesh size of $\Delta x = 2^{-7}$. We use the completed IBDL method with a Fourier spectral discretization and $\eta = 10$. For our implementation, we calculate the derivatives for the nonlinear term after completing the interpolation step of the IBDL method. We may instead choose to calculate these derivatives prior to the interpolation and/or by setting the nonphysical domain solutions equal to 0 to enforce continuity. Initial observations show that these options give comparable results, but more investigation is needed.

In the case of Navier-Stokes, we examine the dimensionless drag coefficient, defined as

$$C_D = \frac{B_1}{u_\infty^2 R} \tag{6.77}$$

where $\boldsymbol{B}$ is the net force on the cylinder. In the case of Stokes flow, we derived equations for the net force on the cylinder using the boundary distributions of $\boldsymbol{F}$ and $\boldsymbol{Q}$ for IBSL and IBDL, respectively. In the case of Navier-Stokes, at steady state, this equation can still be utilized for the IBSL method, but we have found this not to be the case for the IBDL method. More investigation into this matter is needed. Therefore, in the case of Navier-Stokes, we calculate $\boldsymbol{B}$ using a more general formula, which we now derive.

Let us define R as the region inside a box that encircles Γ. Then let us integrate Equation (6.72a) over $R' \equiv R \cap \Omega$. Then, using incompressibility to rewrite the second term and replacing the right-hand side with $\nabla \cdot \boldsymbol{\sigma}$, we get

$$0 = \int_{R'} \rho \boldsymbol{u}_t d\boldsymbol{x} + \int_{R'} \nabla \cdot \boldsymbol{u}\boldsymbol{u}^\mathsf{T} d\boldsymbol{x} - \int_{R'} \nabla \cdot \boldsymbol{\sigma} d\boldsymbol{x}. \tag{6.78}$$

Then, using the divergence theorem and the fact that the boundary of R' is given by $\partial R' = \partial R \cup \Gamma$, we get

$$0 = \int_{R'} \rho \boldsymbol{u}_t d\boldsymbol{x} + \int_{\partial R} (\boldsymbol{u}\boldsymbol{u}^\mathsf{T} - \boldsymbol{\sigma}) \cdot \boldsymbol{n}_R dl(\boldsymbol{x}) + \int_{\Gamma} (\boldsymbol{u}\boldsymbol{u}^\mathsf{T} - \boldsymbol{\sigma}) \cdot \boldsymbol{n} dl(\boldsymbol{x}), \tag{6.79}$$

where $\boldsymbol{n}_R$ and $\boldsymbol{n}$ point out of R'. Then since $\boldsymbol{u}\boldsymbol{u}^\mathsf{T} = 0$ on Γ for flow past a cylinder, we get

$$\boldsymbol{B} = -\int_{\Gamma} \boldsymbol{\sigma} \cdot \boldsymbol{n} dl(\boldsymbol{x}) = -\int_{R'} \rho \boldsymbol{u}_t d\boldsymbol{x} + \int_{\partial R} (\boldsymbol{\sigma} - \boldsymbol{u}\boldsymbol{u}^\mathsf{T}) \cdot \boldsymbol{n}_R dl(\boldsymbol{x}). \tag{6.80}$$

121

(a) IBSL

(b) IBDL

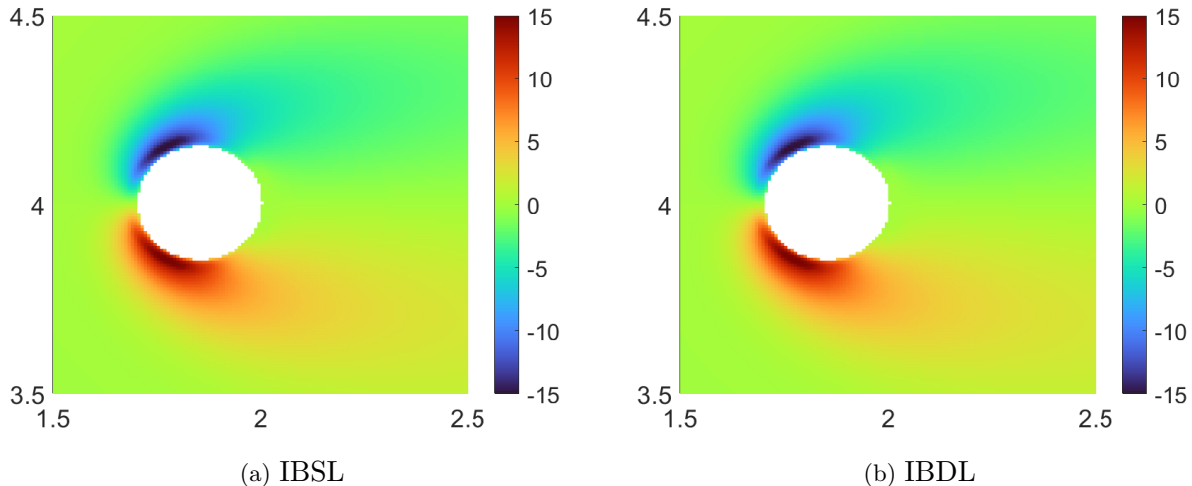FIGURE 6.7. Vorticity plots for Navier-Stokes flow past a cylinder for $Re = 10$ at $t = 144$, using the IBSL and IBDL methods. The computational domain is the periodic box $[0, 8]^2$, the boundary and grid point spacings are $\Delta s \approx \Delta x = 2^{-7}$, the cylinder has radius 0.15. Both methods use a Fourier spectral method.

Drag		
	IBSL	IBDL
C_D	0.3160	0.3137

TABLE 6.4. Dimensionless drag coefficients for $Re = 10$ flow past a cylinder, calcuated with IBSL and IBDL methods. The values are averaged over the time period $t = 54$ to $t = 144$.

In the case of steady flow, the first term vanishes. We use this expression to calculate the force $\boldsymbol{B}$ for Navier-Stokes. In this dissertation, we use the box $R = [0.5, 3.203125] \times [2.8125, 5.1875]$.

6.5.2 Results: $Re = 10$

We first solve the PDE for $Re = 10$ up to time $t = 144$ and compare the results from the IBSL method and the IBDL method, with an interpolation width of $m_1 = 3$ and interior interpolation points $m_2 = 4$ meshwidths from the boundary. Figure 6.7 shows the vorticity plots for the IBSL and IBDL methods, and to the eye, we see good vorticity agreement. Table 6.4 shows the dimensionless drag coefficients for the two methods, and they are within 1% of each other. Further explorations will be made into the affect of the choices of η, interpolation width, and the method for calculation of the nonlinearity.
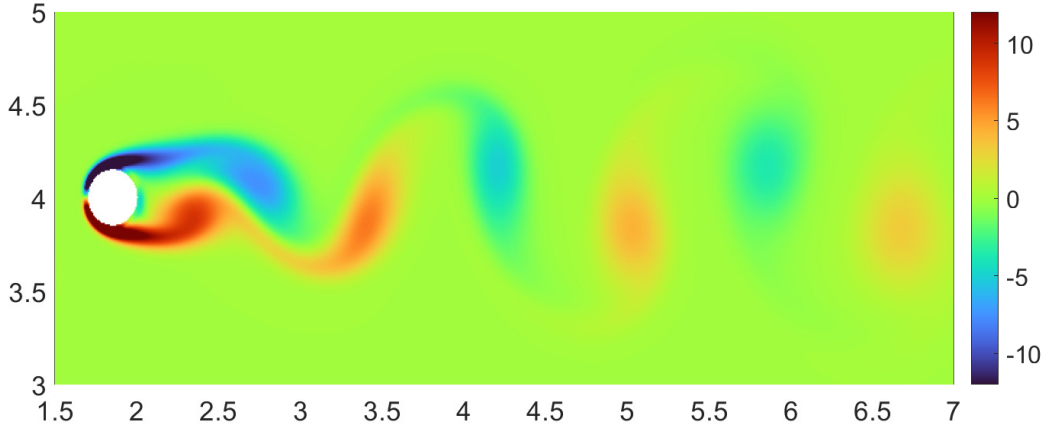
122

FIGURE 6.8. Plot of vorticity for Navier-Stokes flow past a cylinder for $Re = 100$, using the IBDL method with a Fourier spectral discretization. The computational domain is the periodic box $[0, 8]^2$, the boundary and grid point spacings are $\Delta s \approx \Delta x = 2^{-7}$, the cylinder has radius 0.15, and the method uses an interpolation width of $m_1 = 6$ and second interpolation points $m_2 = 8$ meshwidths from the boundary.

6.5.3 Results: $Re = 100$

We next solve the PDE for $Re = 100$ up to time $t = 234$ and compare the results from the IBDL method, with $m_1 = 6$ and $m_2 = 8$, to that of Lai and Peskin [38]. Figure 6.8 shows the vorticity plot for the IBDL method, demonstrating the vortex shedding that we expect with this Reynolds number. We then report the Strouhal number, which gives the dimensionless frequency with which the vortices are shed. It is defined by

$$St = \frac{2Rf_{vs}}{u_\infty}, \tag{6.81}$$

where f_{vs} is the frequency for vortex shedding, which we can measure using $f_{vs} = 1/t_{vs}$, where t_{vs} is the average time between peaks in the lift coefficient. We compare to the Strouhal numbers in [38], found using two different combinations of time steps, mesh sizes and stiffness parameters. We see that the Strouhal number for the IBDL method matches one of the values from [38] to three digits. We do not report drag and lift coefficients as more exploration is needed into the sensitivity of these quantities to various IBDL parameters that may be exaggerated in the case of unsteady flow.

Strouhal Number		
	IBDL	$L\&P_1$	$L\&P_2$
St	0.155	0.155	0.165

TABLE 6.5. Strouhal numbers for $Re = 100$ flow past a cylinder. The first column is calcluated with the IBDL method using a Fourier spectral discretization and an interpolation width of $m_1 = 6$, a time step of $\Delta t = 1.8(10^{-3})$, and a grid point spacing of $\Delta x = 2^{-7}$. The second column gives the results from Scheme 2 in Peskin and Lai [38], using $\Delta t = 1.8(10^{-3})$, $\Delta x = 2^{-6}$, and stiffness $\kappa = 4.8(10^{-4})$. The third column gives the results from Scheme 2, using $\Delta t = 0.9(10^{-3})$ and $\Delta x = 2^{-7}$, and $\kappa = 9.6(10^{-4})$.

Chapter 7

Discussion

We have developed the Immersed Boundary Double Layer method, a numerical method for linear PDE on complex domains with Dirichlet or Neumann boundary conditions that is more efficient than the constraint formulation of the Immersed Boundary method. We achieved this greater efficiency by reformulating the constraint IB method to correspond to a regularized double layer integral equation, which has better conditioning than the single layer integral equation to which the original constraint formulation corresponds. With this better conditioning, we can solve for the Lagrangian dipole force distribution, Q, in a small number of iterations of a Krylov method, and the iteration count does not increase as we refine the mesh. Furthermore, the computed Q is relatively smooth and converges reasonably to the exact potential strength. Both this convergence and the iteration counts are in stark contrast to the IB constraint method, in which the Lagrangian delta force distribution, F, requires many iterations to compute, the number of which increases with finer meshes and tighter boundary point spacing. Additionally, this distribution often fails to converge due to high-frequency noise.

The IBDL method is related to the method of immersed layers, presented in the recent work of Eldredge [17]. In this work, Eldredge uses the indicator function discussed in Section 5.3.5 to develop extended forms of PDEs that govern variables that define different functions on either side of the boundary. Jump quantities corresponding to the strengths of single and double layer potentials naturally emerge in the PDEs. The goal of this method is to be able to enforce different constraints on either side of the immersed boundary in order to obtain solutions on both sides and to accurately predict the surface traction on one side of the boundary. With the inclusion of jumps in both the solution and derivative, this method is able to achieve these goals. We should also note here that our application of the IBDL method to a Neumann problem can be seen as a specific case

125

of the method of immersed layers [17]. However, if we focus on Dirichlet problems, the method of immersed layers again requires inverting an operator corresponding to a first-kind integral equation that is poorly conditioned.

However, the IBDL method is used for Dirichlet problems for which the solution is only desired on one side of the immersed boundary, and it uses a double layer potential whose strength is the unknown jump in solution values. This allows us to develop a method that corresponds to a second-kind integral equation that can be solved much more efficiently. One can see the IB constraint method of Taira and Colonius [59] as corresponding to a single layer integral representation, the IBDL method presented in this dissertation as corresponding to a double layer integral representation, and the method of immersed layers of Eldredge [17] as corresponding to an integral representation that includes both single and double layers. And in the case of Dirichlet boundary conditions, the better conditioning of the IBDL operator gives a more efficient method.

We also make note of another difference between the IBDL method and the method of immersed layers. While [17] uses a lattice Green's function for unbounded external flows, the IBDL method does not involve an explicit Green's function. Instead the method obtains the convolution of the boundary potential with a regularized Green's function by solving the PDE after spreading the boundary potential with the regularized delta function. Thus the IBDL method can be used on more general domains. In this paper we use a periodic computational domain, but one could also easily use other boundary conditions and domains.

The IBDL method retains much of the flexibility and robustness of the IB method. The communication between the Lagrangian coordinate system and the underlying Cartesian grid is achieved with convolutions with regularized delta functions, and no analytical Green's functions are needed in the computation. Furthermore, minimal geometric information is needed for the immersed boundary. We only need boundary points, unit normals, an indicator function flagging points as inside the PDE domain Ω, and an indicator function flagging points as near-boundary points. Furthermore, we can reduce this input list to just the set of points. The unit normals can be estimated as described in Section 5.3.1, and the interior indicator function can be calculated as described in Section 5.3.5. Since such minimal geometric information is needed, this method can be easily used for complex PDE domains. While we focused on two dimensions in this dissertation, nothing in

the method formulation requires this. We chose to use an arclength parametrization in our method development, and in three dimensions, this would simply require that we estimate the areas of the discretized boundary elements. Lastly, we would again require unit normal approximations.

There are a few ways in which the method is not as flexible and easily generalized as the IBSL method. Unlike the original IB constraint method, the IBDL method produces a discontinuous solution across the boundary, requiring careful handling of near-boundary points in order to maintain pointwise convergence in this region. We have found that in the case of a scalar PDE, a finite difference discretization often requires less interpolation. However, a Fourier spectral method can still be utilized by interpolating a region for which the size approaches 0 as we refine the mesh. In the case of Stokes and Brinkman equations, we have found that the form of the IBDL method complicates the implementation of a finite difference method. We have achieved first-order accuracy by approximating the discrete wide Laplacian with the regular Laplacian and utilizing a smoother delta function. However, one area of future research is in developing a more suitable finite difference discretization.

Additionally, we have found that in the case of the exterior Poisson and Stokes equations, the constraint matrix is singular with a one-dimensional nullspace. It therefore requires careful handling of the inversion process. However, by adding a single layer term to form the completed IBDL method, we get a constraint matrix that is invertible, as in the IBSL case. Another area of future research is in better understanding this nullspace and its role in the IBDL method.

The increased efficiency of this method makes it practical for use on time-dependent PDEs. For example, with an implicit time-discretization of the diffusion equation, we could use the IBDL method to solve a Helmholtz equation at each time step. We can also use the IBDL method for a PDE with a nonlinearity by using an implicit-explicit time-stepping scheme, as is done for Navier-Stokes equation in Section 6.5.

Another benefit of the IBDL method is the convergence of the potential strength. This convergence provides the opportunity to use it to obtain the normal derivative of the solution on the boundary in the case of a scalar PDE or boundary traction in the case of Stokes. We would use analytical expressions for derivatives of double layer potentials in order to achieve this. This opens the possibility of improving the IBDL method by obtaining a continuous extension of the solution

127

across the boundary, possibly eliminating the need for the interpolation step of the method. Another area of future work is in conceivably using such normal derivatives on the boundary to create a method like the Immersed Boundary Smooth Extension (IBSE) method [54] to obtain higher accuracy in the IB method. The IBSE method can achieve this higher accuracy, but the poor conditioning of the constraint method is exacerbated, making the method impractical for three dimensions or moving domains. With the greater efficiency of the IBDL method, we may be able to achieve a higher order convergence without this limitation. We therefore have several paths forward for improving the IBDL method, using it to achieve higher accuracy, and applying it to a wider range of problems.

Bibliography

[1] J. F. AHNER, *Some spectral properties of an integral operator in potential theory*, Proceedings of the Edinburgh Mathematical Society, 29 (1986), pp. 405–411.

[2] U. M. ASCHER, S. J. RUUTH, AND B. T. WETTON, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM Journal on Numerical Analysis, 32 (1995), pp. 797–823.

[3] K. E. ATKINSON, *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge University Press, 1997.

[4] Y. BAO, A. DONEV, B. E. GRIFFITH, D. M. MCQUEEN, AND C. S. PESKIN, *An immersed boundary method with divergence-free velocity interpolation and force spreading*, Journal of Computational Physics, 347 (2017), pp. 183–206.

[5] J. T. BEALE AND M. LAI, *A method for computing nearly singular integrals*, SIAM Journal of Numerical Analysis, 38 (2001), pp. 1902–1925.

[6] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[7] R. P. BEYER AND R. LEVEQUE, *Analysis of a one-dimensional model for the immersed boundary method*, SIAM Journal on Numerical Analysis, 29 (1992), pp. 332–364.

[8] A. BHALLA, R. BALE, B. GRIFFITH, AND N. PATANKAR, *A unified mathematical framework and an adaptive numerical method for fluid-structure interaction with rigid, deforming, and elastic bodies*, Journal of Computational Physics, 250 (2013), pp. 446–476.

[9] D. C. BOTTINO, *Modeling viscoelastic networks and cell deformation in the context of the immersed boundary method*, Journal of Computational Physics, 147 (1998), pp. 86–113.

[10] H. C. BRINKMAN, *A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles*, Applied Scientific Research, 1 (1949).

[11] C. CARVALHO, S. KHATRI, AND A. D. KIM, *Asymptotic approximations for the close evaluation of double-layer potentials*, SIAM Journal of Scientific Computation, 42 (2020), pp. A504–A533.

[12] H. D. CENICEROS, J. E. FISHER, AND A. M. ROMA, *Efficient solutions to robust, semi-implicit discretizations of the immersed boundary method*, Journal of Computational Physics, 228 (2009), pp. 7137–7158.

[13] D. COLTON AND R. KRESS, *Integral Equation Methods in Scattering Theory*, John Wiley & Sons, 1983.

[14] R. Cortez, N. Cowen, L. Fauci, and R. Dillion, *Simulation of swimming organisms: coupling internal mechanics with external fluid dynamics*, Computing in Science and Engineering, 6 (2004), pp. 3–45.

[15] R. H. Dillon and L. J. Fauci, *An integrative model of internal axoneme mechanics and external fluid dynamics in ciliary beating*, Journal of Theoretical Biology, 207 (2000), pp. 415–430.

[16] C. D. Eggleton and A. S. Popel, *Large deformation of red blood cell ghosts in a simple shear flow*, Physics of Fluids, 10 (1998), pp. 1834–1845.

[17] J. D. Eldredge, *A method of immersed layers on cartesian grids, with application to incompressible flows*, Journal of Computational Physics, 448 (2022).

[18] V. Flamini, A. DeAnda, and B. E. Griffith, *Immersed boundary-finite element model of fluid-structure interaction in the aortic root*, Theoretical and Computational Fluid Dynamics, 30 (2016), pp. 139–164.

[19] A. L. Fogelson and C. S. Peskin, *A fast numerical method for solving the three-dimensional stokes' equations in the presence of suspended particles*, Journal of Computational Physics, 79 (1988), pp. 50–69.

[20] D. Goldstein, R. Handler, and L. Sirovich, *Modeling a no-slip flow boundary with an external force field*, Journal of Computational Physics, 105 (1993), pp. 354–366.

[21] A. Goza, S. Liska, B. Morley, and T. Colonius, *Accurate computation of surface stresses and forces with immersed boundary methods*, Journal of Computational Physics, 321 (2016), pp. 860–783.

[22] S. Granzier-Nakajima, R. D. Guy, and C. Zhang-Molina, *A numerical study of metachronal propulsion at low to intermediate reynolds numbers*, Fluids, 5 (2020), pp. 2311–5521.

[23] L. Greengard and M. C. Kropinski, *Integral equation methods for stokes flow in doubly-periodic domains*, Journal of Engineering Mathematics, 48 (2004), pp. 157–170.

[24] B. E. Griffith, *Immersed boundary model of aortic heart valve dynamics with physiological driving and loading conditions*, International Journal for Numerical Methods in Biomedical Engineering, 28 (2012), pp. 317–345.

[25] ———, *On the volume conservation of the immersed boundary method*, Communications in Computational Physics, 12 (2012), pp. 401–432.

[26] B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin, *An adaptive, formally second order accurate version of the immersed boundary method*, Journal of Computational Physics, 223 (2007), pp. 10–49.

[27] B. E. Griffith and X. Luo, *Hybrid finite difference/finite element immersed boundary method*, International Journal for Numerical Methods in Biomedical Engineering, 33 (2017), p. e2888.

[28] B. E. Griffith, D. M. McQueen, and C. S. Peskin, *Simulating the fluid dynamics of natural and prosthetic heart valves using hte immersed boundary method*, International Journal of Applied Mechanics, 1 (2009), pp. 137–177.

[29] B. E. Griffith and C. S. Peskin, *On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems*, Journal of Computational Physics, 208 (2005), pp. 75–105.

[30] R. D. Guy, B. Philip, and B. E. Griffith, *Geometric multigrid for an implicit-time immersed boundary method*, Advances in Computational Mathematics, 41 (2015), pp. 635–662.

[31] F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, The Physics of Fluids, 8 (1965), pp. 2182–2189.

[32] H. Hasimoto, *On the periodic fundamental solutions of the stokes equations and their application to viscous flow past a cubic array of spheres*, Journal of Fluid Mechanics, 5 (1959), pp. 317–328.

[33] F.-K. Hebeker, *Efficient boundary element methods for three-dimensional exterior viscous flows*, Numerical Methods for Partial Differential Equations, 2 (1986), pp. 273–297.

[34] G. C. Hsiao and R. Kress, *On an integral equation for the two-dimensional exterior stokes problem*, Applied Numerical Mathematics, 1 (1985), pp. 77–93.

[35] B. Kallemov, A. Bhalla, B. Griffith, and A. Donev, *An immersed boundary method for rigid bodies*, Communications in Applied Mathematics and Computational Science, 11 (2016), pp. 79–141.

[36] Y. Kim and C. S. Peskin, *Penalty immersed boundary method for an elastic boundary with mass*, Physics of Fluids, 19 (2007), pp. 1–18.

[37] A. Klockner, A. Barnett, L. Greengard, and M. O'Neil, *Quadrature by expansion: A new method for the evaluation of layer potentials*, Journal of Computational Physics, 252 (2013), pp. 332–349.

[38] M. Lai and C. S. Peskin, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, Journal of Computational Physics, 160 (2000), pp. 705–719.

[39] S. Lim, A. Ferent, S. Wang, and C. S. Peskin, *Dynamics of a closed rod with twist and bend in fluid*, SIAM Journal on Scientific Computing, 31 (2008), pp. 273–302.

[40] Y. Liu and Y. Mori, *Properties of discrete delta functions and local convergence of the immersed boundary method*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 2986–3015.

[41] D. M. McQueen and C. S. Peskin, *A three-dimensional computer model of the human heart for studying cardiac fluid dynamics*, Computer Graphics, 34 (2000), pp. 56–60.

[42] D. M. McQueen, C. S. Peskin, and E. L. Yellin, *Fluid dynamics of the mitral valve: Physiological aspects of a mathematical model*, American Journal of Physiology, 11 (1982), pp. H1095–H1110.

[43] S. Olson, S. Lim, and R. Cortez, *Modeling the dynamics of an elastic rod with intrinsic curvature and twist using a regularized stokes formulation*, Journal of Computational Physics, 238 (2013), pp. 169–187.

[44] C. S. Peskin, *Flow patterns around heart valves: a digital computer method for solving the equations of motion*, PhD thesis. Albert Einstein Coll. Med. University Microfilms, (1972), pp. 378: 72–30.

[45] ——, *Numerical analysis of blood flow in the heart*, Journal of Computational Physics, 25 (1977), pp. 220–252.

[46] ——, *The fluid dynamics of heart valuves: experimental, theoretical and computational methods*, Annual Review of Fluid Mechanics, 14 (1981), pp. 235–259.

[47] ——, *The immersed boundary method*, Acta Numerica, 11 (2002), pp. 479–517.

131

[48] H. Power, *The completed double layer boundary integral equation method for two-dimensional stokes flow*, IMA Journal of Applied Mathematics, 51 (1993), pp. 123–145.

[49] H. Power and G. Miranda, *Second kind integral equation formulation of stokes' flows past a particle of arbitrary shape*, SIAM Journal on Applied Mathematics, 47 (1987), pp. 689–698.

[50] C. Pozrikidis, *Boundary integral and singularity methods for linearized viscous flow*, Cambridge University Press, 1992.

[51] ———, *A Practical Guide to Boundary Element Methods*, Chapman & Hall, 2002.

[52] A. S. Sangani and . Acrivos, *Slow flow through a periodic array of spheres*, Interntational Journal of Multiphase Flow, 8 (1982), pp. 343–360.

[53] J. Saranen and G. Vainikko, *Periodic Integral and Pseudodifferential Equations with Numerical Approximation*, Springer, 2000.

[54] D. B. Stein, R. D. Guy, and B. 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 (2016), pp. 252–274.

[55] J. M. Stockie, *Modelling and simulation of porous immersed boundaries*, Computers and Structures, 87 (2009), pp. 701–709.

[56] J. M. Stockie and S. I. Green, *Simulating the motion of flexible pulp fibres using the immersed boundary method*, Journal of Computational Physics, 147 (1998), pp. 147–165.

[57] W. Strychalski and R. D. Guy, *Viscoelastic immersed boundary methods for zero reynolds number flow*, Communications in Computational Physics, 12 (2012), pp. 462–478.

[58] S. Su, M. Lai, and C. Lin, *An immersed boundary technique for simulating complex flows with rigid boundary*, Computers and Fluids, 36 (2007), pp. 313–324.

[59] K. Taira and T. Colonius, *The immersed boundary method: a projection approach*, Journal of Computational Physics, 225 (2007), pp. 2118–2137.

[60] J. M. Teran and C. S. Peskin, *Tether force constraints in stokes flow by the immersed boundary method on a periodic domain*, SIAM Journal of Scientific Computing, 31 (2009), pp. 3404–3416.

[61] M. Uhlmann, *An immersed boundary method with direct forcing for the simulation of particulate flows*, Journal of Computational Physics, 209 (2005), pp. 448–476.

[62] F. B. Usabiaga, B. Kallemov, B. Delmotte, A. Bhalla, B. Griffith, and A. Donev, *Hydrodynamics of suspensions of passive and active particles: a rigid multiblob approach*, Communications in Applied Mathematics and Computational Science, 11 (2016), pp. 217–296.

[63] C. Zhang, R. Guy, B. Mulloney, Q. Zhang, and T. Lewis, *Neural mechanism of optimal limb coordination in crustacean swimming*, Proceedings of the National Academy of Sciences of the USA, 38 (2014), pp. 13840–13845.