

Spectrally accurate solutions to inhomogeneous elliptic PDE in smooth geometries using function intension

David Stein^{1,*}

¹*Center for Computational Biology, Flatiron Institute, New York, NY 10010, USA*

We present a spectrally accurate embedded boundary method for solving linear, inhomogeneous, elliptic partial differential equations (PDE) in general smooth geometries, focusing in this manuscript on the Poisson, modified Helmholtz, and Stokes equations. Unlike several recently proposed methods which rely on function extension, we propose a method which instead utilizes function *intension*, or the smooth truncation of known function values. Similar to those methods based on extension, once the inhomogeneity is truncated we may solve the PDE using any of the many simple, fast, and robust solvers that have been developed for regular grids on simple domains. Function intension is inherently stable, as are all steps in the proposed solution method, and can be used on domains which do not readily admit extensions. We pay a price in exchange for improved stability and flexibility: in addition to solving the PDE on the regular domain, we must additionally (1) solve the PDE on a small auxiliary domain that is fitted to the boundary, and (2) ensure consistency of the solution across the interface between this auxiliary domain and the rest of the physical domain. We show how these tasks may be accomplished efficiently (in both the asymptotic and practical sense), and compare convergence to several recent high-order embedded boundary schemes.

I. INTRODUCTION

Let \mathcal{L} denote a constant coefficient elliptic operator, Ω a simply connected compact subset of \mathbb{R}^2 with smooth boundary Γ , and \mathcal{B} a boundary operator (e.g. $\mathcal{B}u = u$ for Dirichlet boundary conditions). We assume that the specified inhomogeneity f is smooth in Ω and the specified boundary condition g is smooth on Γ . We seek to find a solution u to the partial differential equation (PDE)

$$\begin{aligned} \mathcal{L}u &= f && \text{in } \Omega, && (1a) \\ \mathcal{B}u &= g && \text{on } \Gamma. && (1b) \end{aligned}$$

There are two cases where optimal methods to this problem exist. The first of these is when Ω is geometrically simple: if Ω is the doubly periodic rectangle \mathbb{T}^2 , spectral methods based on the Fast-Fourier transform (FFT) provide an optimal method for the inversion of \mathcal{L} ; similar methods exist for several other simple geometries based on different spectral expansions [1] or adaptive integration using quadtrees [2]. The second case is when Ω need not be geometrically simple, but $f = 0$. In this case, well-conditioned boundary integral equation (BIE) methods exist for many commonly studied operators \mathcal{L} ; spectrally accurate or high-order singular quadratures for the associated Nystrom schemes along with kernel-dependent and kernel-independent Fast Multipole Methods (FMMs) enable accurate solutions to be computed and evaluated in optimal time [3–14].

When the domain is complicated, as in the domain shown in Figures 1 and 2, and the problem is inhomogeneous (i.e. $f \neq 0$) the situation is less clear. While methods have been continuously improving over the last several decades, all methods lack optimality in some way: whether due to slow convergence; ill-conditioning; or long compute times. Recently, considerable interest has focused on methods utilizing function extension, where either the inhomogeneity or the unknown solution defined on the general domains Ω is extended beyond its known values. These works include active penalty methods [15], variations on the Immersed Boundary method [16, 17], methods utilizing radial basis functions (PUX) [18–20], Fourier continuation methods [21–24], and those relying entirely on BIE [25]. Although promising, as these methods can produce relatively high-order discretizations with reasonable compute times, function extension is an inherently ill-conditioned process, as evinced by the great pains that some of these methods have taken to provide stability, and we worry that such methods will have issues both when steep boundary layers arise (as has been our own experience in the simulation of complex fluids [26] and dissolution problems [27]), and when domains curve back on themselves.

We provide an alternative embedded boundary scheme that shares many of the benefits of those methods that utilize function extension, while eliminating some of the drawbacks. Rather than attempting to extend the inhomogeneity,

* dstein@flatironinstitute.org

we smoothly roll it off to 0 *inside of the domain*. Because this does not require extrapolation, it is both inherently *stable* and relatively simple, requiring only the distance to the boundary and a regularized cutoff function. We then solve the PDE with this modified right-hand side using a regular grid method. Unfortunately, the story doesn't end there, as this candidate solution does not satisfy the PDE in the entire domain Ω . To correct errors near to the boundary, we solve an *annular* problem in a thin boundary-fitted annulus. The discrepancy between the annular solution and the solution on the regular grid is corrected, and boundary conditions are finally enforced by solving a homogeneous PDE utilizing a well-conditioned BIE method.

In this manuscript we present both the abstract method and a relatively simple implementation which makes use of global discretizations for both the boundary and regular grid. For problems that are not significantly multiscale in nature, this turns out to be reasonably performant, with virtually all steps having asymptotic scalings that are *less* than the FFTs used to solve the regular grid problem. For simplicity, we discretize certain steps using methods that depend on dense linear algebra which have slightly worse scaling in the *setup* stage of the problem, but not in the *solution* stage of the problem; meaning that the overall scaling of solving repeated problems on the same geometry is *the same as solving the regular PDE on a periodic grid* (albeit with worse constants). We make comments where poorly scaling methods are utilized, and how they could be improved upon (using methods already available in the literature); but in practice, these stages are rarely limiting for moderately sized problems as they make use of efficient BLAS and LAPACK routines.

This paper is organized as follows. In Section II, we introduce the basic methodology of *function intension* for solving PDE of the form given in Equation (1). This presentation will be simple and stripped down, both for pedagogical purposes and to emphasize the modular nature of the abstract method, where many of the substages have wide freedom of implementation, with little detailed interdependence. Then in Section III, we introduce some basic preliminaries which will make discussion of our specific implementation easier. In Section IV, we revisit our abstract presentation of the method, now providing details for the specific implementational choices made throughout this paper. In Section V, we explore how the method can be used to generate a solver of fixed algebraic order for any $M > 0$, and show how that order can naturally be varied with the discretization to produce a spectrally accurate scheme. In Section VI, we discuss how to set the various parameters required by the method, and with these fixed, demonstrate large N stability for the simple problem studied in the previous section. For this same problem, we show practical wall-clock timings across a wide range of problem sizes. To wrap up the presentation of the methodology, in Section VII we provide algorithms for both the setup and solve components of our solver, with asymptotic scalings given for all compute-heavy steps.

We then turn to results for more complex problems set on more difficult domains. In Section VIII, we extend our method to multiply-connected domains, and compare our Poisson solver to the Partition of Unity Extension (PUX) method [18]. In Section IX, we compare our method to the two-dimensional Fourier continuation method [23], using this comparison to emphasize some differences between function intension and function extension. In Section X, we discretize and solve a modified Helmholtz problem, with a large parameter α as arises when time-discretizing the diffusion equation. We again compare our results to those produced by the PUX method [19], finding very close agreement in the rate of convergence between the two methods, and similar stability as α grows. Finally, we solve a Stokes problem, comparing both errors and run-times with the IBSE method [17]. In Section XII, we conclude by discussing some of the outstanding issues with the method proposed here, and discuss some possible improvements to the method, from both a methodological and practical point of view.

II. METHODS — OVERVIEW

To avoid the main ideas being lost in a thicket of details, we begin with a highly simplified discussion of the overall method, deferring most questions of implementation to Sections III and IV. For further simplicity, let us choose as a specific case the interior Poisson problem with Dirichlet boundary conditions:

$$\Delta u = f \quad \text{in } \Omega, \quad (2a)$$

$$u = g \quad \text{on } \Gamma, \quad (2b)$$

with \mathcal{L} and \mathcal{B} from Equation (1) given by the Laplace operator and the interior trace operator for the boundary Γ , respectively. We begin by specifying a domain decomposition and defining several regions of space, boundaries, and interfaces, all shown in Figure 1. The domain Ω on which the PDE is defined will be referred to as the *physical domain*, with a boundary $\Gamma = \partial\Omega$. A thin-strip region \mathcal{A} , referred to as the *annular domain*, is defined along the boundary Γ , within Ω . The physical region $\tilde{\Omega} = \Omega \setminus \mathcal{A}$ that is far from the boundary will be referred to as the *faithful domain*. The curve \mathcal{I} that separates \mathcal{A} and $\tilde{\Omega}$ will be referred to as the *interface*. Finally, we assume that Ω is contained within a geometrically simple domain \mathcal{C} , referred to as the *computational domain*. Our method makes use of these spaces to

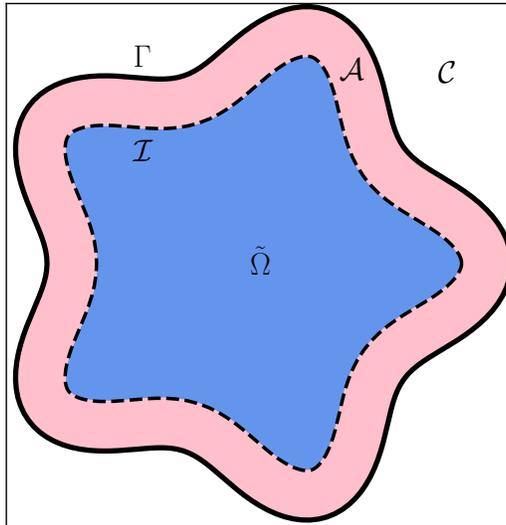


FIG. 1: An example of a domain Ω (everything inside the solid black curve Γ , and the union of the pink and blue regions), embedded into a geometrically simple computational domain \mathcal{C} (everything inside the rectangle that contains Ω). The domain Ω is decomposed into the faithful domain $\tilde{\Omega}$, shown in blue, and the annular domain \mathcal{A} , shown in pink. The faithful domain $\tilde{\Omega}$ and the annular domain \mathcal{A} are separated by an interface \mathcal{I} , denoted by the dashed black line.

perform a simple and straightforward domain decomposition strategy, which is illustrated in Figure 2, and described in the steps below:

1. The function f , known only inside Ω (shown in Figure 2a), is smoothly truncated, so that it is 0 at Γ , unchanged within the faithful domain $\tilde{\Omega}$ and altered only within the thin boundary-adjacent strip \mathcal{A} . We refer to this process as *function intension*. The truncated (or *intended*) function is shown in Figure 2b, and the function in the annular region \mathcal{A} is shown in Figure 2c.
2. The *regular problem* is solved in a geometrically simple region $\mathcal{C} \supset \Omega$ using standard methods, generating a solution \tilde{u} (shown in Figure 2d) that satisfies $\Delta \tilde{u} = f$ in $\tilde{\Omega}$.
3. The *annular problem* is solved in \mathcal{A} , generating a solution $u_{\mathcal{A}}$ (shown in Figure 2e) that satisfies $\Delta u_{\mathcal{A}} = f$ in \mathcal{A} .
4. A *stitching problem* is solved to correct any mismatch at the interface \mathcal{I} between \tilde{u} and $u_{\mathcal{A}}$, generating a single solution u_I that satisfies $\Delta u_I = f$ everywhere in Ω . The “unstitched” function given by \tilde{u} for $\mathbf{x} \in \tilde{\Omega}$ and $u_{\mathcal{A}}$ for $\mathbf{x} \in \mathcal{A}$ is shown in Figure 2f, and the “stitched” function u_I , where the interface mismatch has been corrected, is shown in Figure 2g.
5. A homogeneous boundary-value problem is solved, giving a correction function u_H which satisfies $\Delta u_H = 0$ everywhere in Ω , with boundary conditions chosen so that $u = u_I + u_H$ satisfies the physical boundary conditions $u = g$. The homogeneous correction u_H is shown in Figure 2h, and the solution to the full PDE $u = u_I + u_H$ is shown in Figure 2i.

A. Function intension, and solving the regular problem

The primary goal of this paper is to replicate the main benefit of function extension methods — converting a problem on a complex domain to a problem on a simple domain — without having to extend either the inhomogeneous forcing f or the unknown solution u . Instead, the function is simply cutoff in the interior of the domain Ω , a process we refer to as *function intension*, which we lay out here. The basic definition is simple:

$$f_I(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \mathbf{x} \in \tilde{\Omega}, \\ \eta(\mathbf{x})f(\mathbf{x}), & \mathbf{x} \in \mathcal{A}, \\ 0, & \mathbf{x} \in \Omega^C, \end{cases} \quad (3)$$

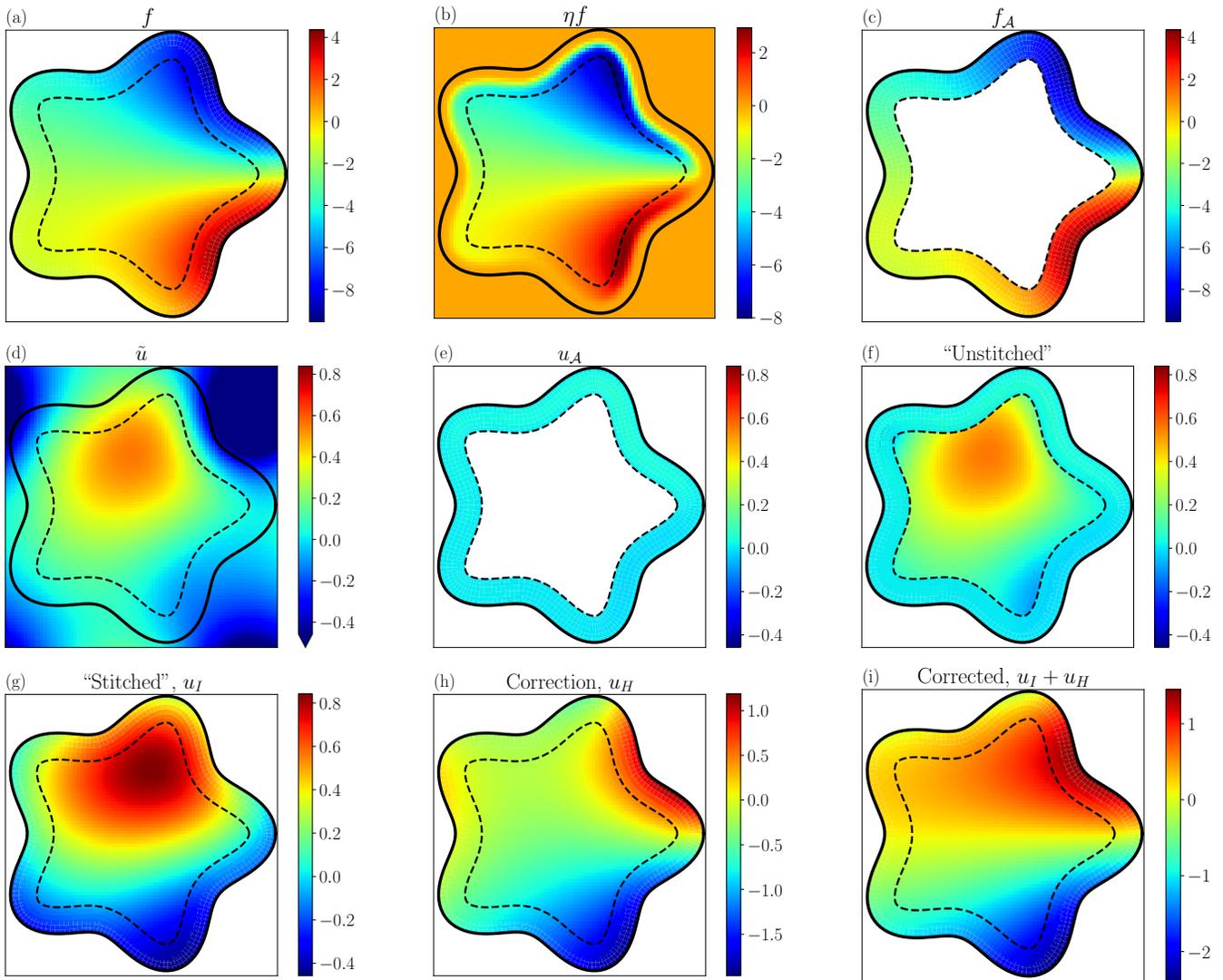


FIG. 2: Schematic of solution process. The inhomogeneous forcing f (panel a) is stably decomposed into a truncated function ηf (panel b) and an annular function f_A (panel c), via the process of *function intension* (see Section II A). Note that the function $\eta f = f$ everywhere inside the faithful domain $\tilde{\Omega}$. The regular solution \tilde{u} is obtained via standard methods on the simple domain \mathcal{C} (panel d), while a thin-strip problem (see Section II B) is solved to obtain u_A (panel e). The “unstitched solution” (panel f) defined by truncating the regular solution \tilde{u} to the faithful domain $\tilde{\Omega}$, and by the annular solution u_A inside \mathcal{A} , is clearly not smooth. Adding layer potentials, as described in Section II C, fixes these discontinuities and produces the “stitched”, or inhomogeneous solution u_I (panel g). Correcting the boundary condition by solving a homogeneous problem (shown in panel h, see Section II D) gives the estimated solution to the problem (panel i).

with η a smooth cutoff function that is 1 at \mathcal{I} and 0 at Γ , although it is simpler to think of f_I being defined simply as $f_I(\mathbf{x}) = \eta(\mathbf{x})f(\mathbf{x})$, with η understood to be 1 for $\mathbf{x} \in \tilde{\Omega}$ and 0 outside of Ω , so that it is irrelevant that f is unknown in Ω^C . The convergence of our scheme will depend asymptotically on the smoothness of η and practically (in terms of constants) on the specific choice of the function, which we make precise in Section III A 4. Because η is smooth, $f_I = \eta f$ is smooth in \mathcal{C} , and so Δ may be inverted using any appropriate method on the geometrically simple domain \mathcal{C} . Let us thus define \tilde{u} by the solution to:

$$\Delta \tilde{u} = \eta f \quad \text{in } \mathcal{C}, \quad (4)$$

along with any appropriate far-field boundary conditions that are required. We refer to \tilde{u} as the *regular problem*. We note that the solution \tilde{u} is not unique and depends on the far-field boundary conditions chosen for \mathcal{C} and any other

modifications that must be made (see Section III B 3).

How good of a guess is \tilde{u} to the actual solution to Equation (1)? Well, it is correct, up to numerical errors, within the faithful domain $\tilde{\Omega}$, i.e. $\Delta\tilde{u}(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x} \in \tilde{\Omega}$. However, \tilde{u} fails to satisfy both the inhomogeneity in \mathcal{A} and fails to satisfy the boundary conditions.

B. The annular problem

The region of Ω in which \tilde{u} fails to satisfy $\Delta\tilde{u} = f$ is, by construction, the annular region \mathcal{A} . We define a second problem, referred to as the *annular problem*, stated here:

$$\Delta u_{\mathcal{A}} = f \quad \text{in } \mathcal{A}, \quad (5a)$$

$$u_{\mathcal{A}} = 0 \quad \text{on } \Gamma \text{ and on } \mathcal{I}. \quad (5b)$$

Solving this problem is nontrivial, but it is also tractable as \mathcal{A} is reasonably geometrically simple. To do so we will make use of a body-fitted coordinate system defined only within \mathcal{A} , but defer discussion of the details to Section IV B. As with \tilde{u} , the solution $u_{\mathcal{A}}$ is not unique, as any boundary conditions which make the PDE well-posed can be chosen. It is clear that $u_{\mathcal{A}}$ satisfies the inhomogeneity in \mathcal{A} — precisely where the regular solution \tilde{u} fails to. It is tempting to define a solution candidate u by \tilde{u} for $\mathbf{x} \in \tilde{\Omega}$ and $u_{\mathcal{A}}$ for $\mathbf{x} \in \mathcal{A}$. There are two issues with this: u will still fail to satisfy the boundary condition given in Equation (1), but more importantly, u may have jumps in both its value and its normal derivative at the interface \mathcal{I} . We consider this more pressing problem first.

C. The stitching problem

At this point, we assume that we have access to the regular solution \tilde{u} for all $\mathbf{x} \in \tilde{\Omega}$, and the annular solution $u_{\mathcal{A}}$ for all $\mathbf{x} \in \mathcal{A}$, and so are free to interpolate and differentiate. We may thus evaluate the jump in both the solution and its normal derivative at the interface:

$$\gamma = \lim_{\mathbf{x} \rightarrow \mathcal{I}^{\tilde{\Omega}}} \tilde{u} - \lim_{\mathbf{x} \rightarrow \mathcal{I}^{\mathcal{A}}} u_{\mathcal{A}}, \quad (6a)$$

$$\sigma = \lim_{\mathbf{x} \rightarrow \mathcal{I}^{\tilde{\Omega}}} \partial_{\mathbf{n}} \tilde{u} - \lim_{\mathbf{x} \rightarrow \mathcal{I}^{\mathcal{A}}} \partial_{\mathbf{n}} u_{\mathcal{A}}, \quad (6b)$$

with $\partial_{\mathbf{n}} f$ denoting the normal derivative of f . For a given elliptic PDE with known jump conditions [3, 4], these discontinuities in the value and the normal derivative of the function may be corrected by adding appropriate layer potentials. This leads to the *inhomogeneous solution*, which for the Poisson equation, takes the form:

$$u_I(\mathbf{x}) = \begin{cases} \tilde{u}(\mathbf{x}) + (\mathcal{S}_{\mathcal{I}}\sigma)(\mathbf{x}) - (\mathcal{D}_{\mathcal{I}}\gamma)(\mathbf{x}) & \text{for } \mathbf{x} \in \tilde{\Omega}, \\ u_{\mathcal{A}}(\mathbf{x}) + (\mathcal{S}_{\mathcal{I}}\sigma)(\mathbf{x}) - (\mathcal{D}_{\mathcal{I}}\gamma)(\mathbf{x}) & \text{for } \mathbf{x} \in \mathcal{A}, \end{cases} \quad (7)$$

where \mathcal{S} and \mathcal{D} denote the single and double layer potential operators associated with the Laplace operator [4]. The inhomogeneous solution u_I now both satisfies the PDE $\Delta u = f$ everywhere in Ω , is continuous, and has a continuous first-derivative, and thus is as smooth as supported by f — in particular, if $f \in C^k(\Omega)$, then $u \in C^{k+2}(\Omega)$ [28]. It is still the case, unfortunately, that u_I fails to satisfy the boundary condition on the original PDE.

D. The homogeneous problem

Finally, we measure how much u_I fails to satisfy the boundary condition by evaluating the discrepancy $\delta_{\Gamma} = g - u_I|_{\Gamma}$. It then remains only to solve the now *homogeneous* equation:

$$\Delta u_H = 0, \quad \text{in } \Omega, \quad (8a)$$

$$u_H = \delta_{\Gamma}, \quad \text{on } \Gamma. \quad (8b)$$

Well-conditioned methods based on boundary-integral equations allow for the solution and fast evaluation for a wide class of common PDE. Finally, having solved for u_H , we may define the solution $u = u_I + u_H$ to Equation (1), valid for all $\mathbf{x} \in \Omega$.

III. METHODS — PRELIMINARIES

Having presented a sketch of the solution process, we now turn to the details of our specific implementation. In this initial manuscript, we focus on a simple global implementation, which nevertheless provides spectral accuracy, along with both setup and solution times with reasonable asymptotic scalings and constants. We begin first by explicitly defining our discretization of the domain Ω , and its decomposition into $\tilde{\Omega}$ and \mathcal{A} .

A. Domain decomposition and discretization

We assume that the boundary curve Γ is given to us as a closed parametrized curve $\mathbf{X}(s) = (X(s), Y(s))$, for $s \in [0, 2\pi)$, with $\mathbf{X}(0) = \mathbf{X}(2\pi)$, with a counter-clockwise orientation. The speed ϕ of this parametrization is given by $\phi(s) = \sqrt{\mathbf{X}_s \cdot \mathbf{X}_s}$. The outward pointing unit normal vector \mathbf{n} is defined by $\phi(s)\mathbf{n}(s) = Y_s(s)\hat{x} - X_s(s)\hat{y}$. Given Γ , we may now define the annular domain \mathcal{A} .

1. Definition of the annular domain \mathcal{A}

Near to the curve Γ , we can define a coordinate system (s, r) by $\mathbf{x} = \mathbf{X}(s) + r\mathbf{n}(s)$, and the annular domain \mathcal{A} is defined to be all points $\mathbf{x} \in \Omega$ with $|r| < R$ for some annular radius R . In the example given in Figure 1, the physical domain Ω is interior to Γ ; for such domains the annular region is defined for $-R \leq r \leq 0$; for domains exterior to Γ the annular region is defined for $0 \leq r \leq R$.

It is clear that R could be chosen so large that the coordinates (s, r) are no longer well defined for all $\mathbf{x} \in \mathcal{A}$. The Jacobian of the coordinate map is given by $\mathcal{J}(\mathbf{s}) = |\partial\mathbf{x}/\partial\mathbf{s}| = \phi(s)(1 + r\kappa(s))$, where the coordinate $\mathbf{s} = (s, r)$ and κ is the signed curvature given by $\kappa = -\mathbf{n} \cdot \mathbf{X}_{ss}/\phi^3$. Let $\kappa = \kappa^+ + \kappa^-$ give the decomposition into the positive (κ^+) and negative (κ^-) parts of κ . For an interior problem, $r < 0$ for $\mathbf{x} \in \mathcal{A}$, and so $R_{\max} = 1/\max \kappa_+$ is sufficient to ensure $\mathcal{J} > 0$. For an exterior problem, $r > 0$, and so $R_{\max} = -1/\min \kappa_-$. More simply, the maximum width of our annular region should be the minimum radius of curvature of Γ , being careful with signs. Because this criteria does not depend on ϕ , the choice of R is purely geometric, and is independent of the parametrization of Γ .

Remark 1 (Coordinate splay). *For our method to work, coordinate collapse must be prevented by choosing $R < R_{\max}$. However, coordinate splay can also occur when R is large. Although this doesn't lead to the breakdown of the method, large coordinate splay leads to the amplification of effective discrete units, so that the discrete annulus provides an inefficient discretization of space.*

Once R is chosen, the interface \mathcal{I} is defined by the parametrized curve $\mathbf{X}(s) \pm R\mathbf{n}(s)$ for $s \in [0, 2\pi)$, with positive sign for exterior problems and negative for interior problems. So long as $R < R_{\max}$, \mathcal{I} has the same regularity as Γ .

2. Discretization of the boundary and annular domain

We take an N point discretization to Γ , with the discrete nodes of the boundary given by $\mathbf{X}_j = \mathbf{X}(s_j)$, where $s_j = j\Delta s$, with $\Delta s = 2\pi/N$. Because \mathbf{X} is periodic, boundary quantities such as \mathbf{n} and ϕ can be computed using FFTs. For an interior problem, the (s, r) rectangle $[0, 2\pi] \times [-R, 0]$ is discretized using an $N \times M$ tensor Fourier/Chebyshev mesh, with discrete values of r at the first-kind Chebyshev points $r_k = -R(\cos(\pi(2k+1)/(2M)) + 1)/2$, for $k = 0, 1, \dots, M-1$, and discrete values of s at the same nodes s_j as used to discretize the boundary. The physical nodes for the discretized annular domain are then $\mathbf{x}_{jk} = \mathbf{X}_j + r_k\mathbf{n}_j$. Modification to the exterior case is straightforward. We define the smallest and largest discrete grid-spacing associated with the discretization of Γ to be $h_{\min} = \min_j \phi_j \Delta s$ and $h_{\max} = \max_j \phi_j \Delta s$, respectively.

3. Definition and discretization of the computational domain \mathcal{C}

We choose \mathcal{C} to be a rectangle, and discretize that rectangle using a simple tensor-product Fourier representation. In principle, and in contrast to methods utilizing function extension, the computational domain \mathcal{C} can be chosen to be tight around the boundary Γ . It is, however, simpler to compute $\min_j (X_j)$ than $\inf_s (X(s))$, and so we define $X_{\min} = \min_j X_j - h_{\max}$, $X_{\max} = \max_j X_j + h_{\max}$, and Y_{\min} and Y_{\max} , analogously. We then define $\mathcal{C} = [X_{\min}, X_{\max} + w] \times [Y_{\min}, Y_{\max} + w]$, where w is a specified amount of wiggle-room. For many PDEs, w can be taken

to be 0, but for Poisson and Stokes type-problems, satisfaction of compatibility conditions for the regular solution may require, in the worst case, $w = 2Mh_{\max}$, see Section III B 3.

Finally, we choose N_x and N_y discrete modes, and adjust the definition of \mathcal{C} , to ensure that we have an isotropic discretization with an even number of modes in each direction. To be precise, we choose $N_x = 2[(X_{\max} + w - X_{\min})/(2h)]$, where $\lceil \cdot \rceil$ denotes the ceiling function and h gives the target resolution of the regular grid (see Section VI). Defining N_y analogously, we finally modify \mathcal{C} to be $\mathcal{C} = [X_{\min}, X_{\min} + hN_x] \times [Y_{\min}, Y_{\min} + hN_y]$.

4. Definition of the cutoff function η

Let us assume for the moment that we have a discrete, scalar valued function H of one variable that smoothly approximates a Heaviside function, with $H(x) = 0$ for all $x < 0$ and $H(x) = 1$ for all $x > 1$. For an interior problem, we can now define a function $\eta(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{C}$ by:

$$\eta(\mathbf{x}) = \begin{cases} 0, & \text{for } \mathbf{x} \in \Omega^C, \\ H(-r(\mathbf{x})/R) & \text{for } \mathbf{x} \in \mathcal{A}, \\ 1, & \text{for } \mathbf{x} \in \tilde{\Omega}, \end{cases} \quad (9)$$

where $r(\mathbf{x})$ gives the r -coordinate for the point $\mathbf{x} \in \mathcal{A}$ (with the obvious modification to be made for exterior problems). It is thus left to define H . While any smooth approximation of the Heaviside function should work, optimizing error requires that the function is both well resolved by the discretization and has continuous derivatives at $x = 0$ and $x = 1$. Throughout this work, we will use an integral of a prolate-spheroidal wavefunction, motivated by recent work on the non-uniform FFT [29]; we have found empirically that this improves performance slightly relative to simpler choices (such as a rescaled error function). To be precise, we use the standard definition of the Digital Prolate Spheroidal Sequence (DPSS), which is given by the dominant eigenvector of a matrix constructed from the sampled sinc function, with bandwidth $b/4$, as implemented in the `scipy.window.dpss` [30, 31]. We sample this discrete bump function with sufficiently high frequency to ensure reconstruction to 15 digits using quintic spline interpolation; and construct H as its antiderivative using adaptive quadrature (again, to 15 digits). This gives bump and step functions defined on $x \in [-1, 1]$ (as shown in Figure 3), which are transformed to the interval $[0, 1]$ via an affine transformation.

Remark 2 (Fast evaluation of DPSS Bump and Step Functions). *To allow fast usage, we have tabulated Chebyshev coefficients for both the bump and step functions for all b such that $b = 1, 2, \dots, 200$. The evaluation of these coefficient expansions can then be further accelerated by exploiting that (once properly centered) the bump functions are even and the step functions are odd [32].*

Examples of these bump and step functions, for various values of b , are shown in Figure 3. The choice of b is considered in Section VI C, but in general higher values of b will be used when M is larger. Naively, it may appear that the bump and step functions for small values of b , such as $b = 4$ are poor choices — they are not even continuous at -1 and 1 . However, these are typically used when M is very small — forcing a strong tradeoff between discrete resolution of the step function and its continuity and smoothness at the boundary.

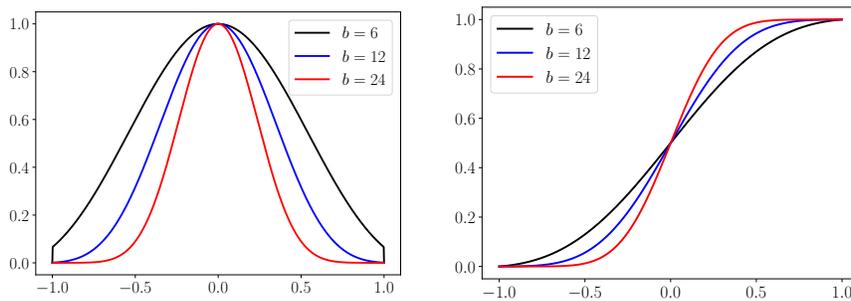


FIG. 3: Regularized step and heaviside functions with bandwidths $b = 6$ (black), $b = 12$ (blue), and $b = 24$ (red).

B. Common subproblems

There are several steps in the algorithm which, while not an intimate part of the solution method, are required and not completely trivial. We discuss these here, to allow a more readable description of the specific implementation

later in Section IV.

1. The physical and coordinate problems

At this point, we have two separate discretizations: one for the simple computational rectangle \mathcal{C} and one for the annular region \mathcal{A} . We must now connect these discretizations, which amounts to being able to accomplish the following task: given $\mathbf{x} \in \mathcal{C}$, determine if

1. Is \mathbf{x} interior or exterior to Ω ?
2. If \mathbf{x} is interior to Ω , is it in \mathcal{A} or $\tilde{\Omega}$?
3. If $\mathbf{x} \in \mathcal{A}$, for what values of (r, s) do we have $\mathbf{x} = \mathbf{X}(s) + r\mathbf{n}(s)$?

We would like to reduce questions (1) and (2) to questions about polygons, for which well-known algorithms [33] with robust implementations [34] can be used. Unfortunately, it is not always the case that points inside the discrete polygon formed by connecting the points Γ_j lie inside Γ . Instead, we define a modified discrete boundary curve $\tilde{\Gamma}$ by $\tilde{\mathbf{X}} = \mathbf{X} + 2\delta\mathbf{n}$, with $\delta = \max_j \delta_j$ and $\delta_j = |\kappa_j^{-1}| - \sqrt{\kappa_j^{-2} - (\Delta s \phi_j / 2)^2}$. The distance δ_j comes from locally approximating the curve by a circle with radius κ_j^{-1} , with a safety buffer of 2 used. A modified discrete interface $\tilde{\mathcal{I}}$ is defined analogously. In Figure 4, we show an example with discrete and continuous curves Γ and \mathcal{I} , along with the modified discrete curves $\tilde{\Gamma}$ and $\tilde{\mathcal{I}}$. This now provides a simple algorithm for classifying \mathbf{x} , which we give for interior problems

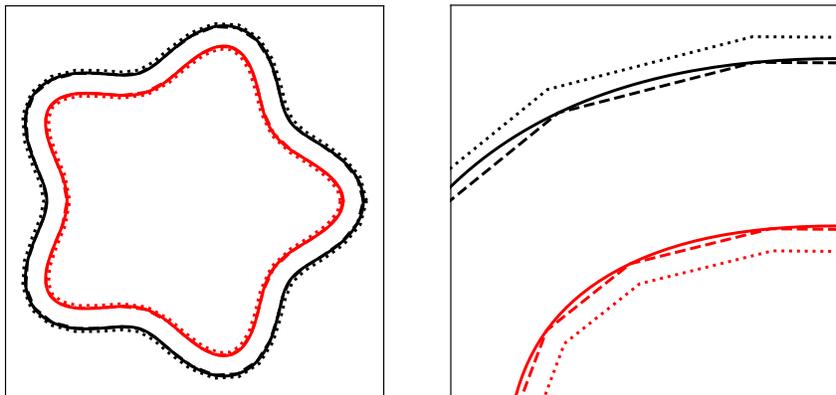


FIG. 4: Discrete (dashed lines), continuous (solid lines), and modified discrete (dotted lines) curves for the boundary Γ (in black) and \mathcal{I} (in red).

(with clear modifications for exterior problems):

Algorithm 1 Classification of \mathbf{x}

$\mathcal{O}(\log N)$

Require: Discrete polygons \mathcal{I} , $\tilde{\mathcal{I}}$, Γ , $\tilde{\Gamma}$, point $\mathbf{x} \in \mathcal{C}$.

- 1: **if** \mathbf{x} is exterior to $\tilde{\Gamma}$ **then return** $\mathbf{x} \in \Omega^C$
 - 2: **else if** \mathbf{x} is interior to $\tilde{\mathcal{I}}$ **then return** $\mathbf{x} \in \tilde{\Omega}$
 - 3: **else**
 - 4: Use Newton's method to solve for the coordinates (s, r) which give $\mathbf{x} = \mathbf{X}(s) + r\mathbf{n}(s)$ ¹.
 - 5: **if** $r > 0$ **then return** $\mathbf{x} \in \Omega^C$
 - 6: **else if** $-R < r < 0$ **then return** $\mathbf{x} \in \mathcal{A}$
 - 7: **else return** $\mathbf{x} \in \tilde{\Omega}$
-

Since N^2 points must be categorized, each step must be carefully optimized. A simple quadtree based approach, where whole blocks can be categorized as either in the exterior Ω^C or faithful domain $\tilde{\Omega}$ is sufficient to reduce the

¹ For a unique solution to this Newton problem to be guaranteed, we require $R + 2\delta < R_{\max}$. While we do check this condition, it fails only when both R is very close to R_{\max} and the boundary discretization is extremely coarse.

lookups (steps 1 and 2) to $\mathcal{O}(\log N)$ for each point \mathbf{x} . Similarly, for a given point \mathbf{x} whose coordinates must be determined, the Newton solver is $\mathcal{O}(1)$, so long as the routines used to interpolate \mathbf{X} and \mathbf{n} are. This can be accomplished, to spectral accuracy, using a type-2 nonuniform FFT, see Remark 3. Finally, explicitly solving the 2D Newton problem $\mathbf{x} = \mathbf{X}(s) + r\mathbf{n}(s)$ is more expensive than solving the scalar minimization problem $\inf_s |\mathbf{X}(s) - \mathbf{x}|$; once s is known r is easily recovered.

2. Interpolation between domains, boundaries, and interfaces

In multiple places in our algorithm, we will be required to interpolate between domains, or between domains and boundaries/interfaces. We discuss a few cases here: (1) Interpolation of a smooth function known everywhere in \mathcal{C} to anywhere else in \mathcal{C} ; (2) interpolation of a function known on the discrete nodes of the annular domain to arbitrary $\mathbf{x} \in \mathcal{A}$; and (3) interpolation of a function known at the discrete nodes of the annular domain to the discrete boundary and interface nodes.

Interpolation from \mathcal{C} to general \mathbf{x} (1) is the simplest. Given a function f on the discrete nodes of \mathcal{C} , we compute its two-dimensional FFT \hat{f} . We may then, after suitably rescaling \mathbf{x} to the unit-periodic rectangle $[0, 2\pi]^2$, compute $f(\mathbf{x})$ by evaluating the type-2 nonuniform FFT (NUFFT). The error in this interpolation converges spectrally fast in the number of modes N_x and N_y used to discretize \mathcal{C} , so long as f is smooth and periodic.

Remark 3 (Accelerating interpolation via NUFFT). *In our algorithm, we will typically have to compute the NUFFT to only $\mathcal{O}(MN)$ targets, with $M \ll N$. Since there are a small number of targets, the interpolation cost is small relative to the oversampled FFT, and we can exploit the option provided in FINUFFT [29] to use an oversampling factor of 1.25 rather than 2. This reduces the cost from approximately 4 times a grid-sized FFT time to ≈ 1.5 times. An additional optimization comes from realizing that only the final interpolation step of the NUFFT algorithm depends on \mathbf{x} ; we may thus precompute all other portions of the algorithm and interpolate to multiple different collections of target values \mathbf{x} at a fixed cost per target \mathbf{x} .*

Interpolating a function f from the discrete nodes of \mathcal{A} to any $\mathbf{x} \in \mathcal{A}$ (2) is only slightly harder. Given an arbitrary $\mathbf{x} \in \mathcal{A}$, we first compute its coordinates (s, r) (see Section IIIB 1). The function f is periodic in s but not in r ; but it is defined on a Chebyshev mesh in r . We may thus take its even reflection and modify it to obtain a periodic function [35], and then, upon suitably rescaling r to lie within the unit periodic interval $[0, 2\pi]$, again use a type-2 NUFFT to compute $f(\mathbf{x})$. Errors in this interpolation converge spectrally fast in the number of boundary modes N and the number of Chebyshev modes M .

Finally, we consider case (3), interpolating a function f from the discrete nodes of \mathcal{A} to either the discrete boundary nodes Γ_j or the discrete interface nodes \mathcal{I}_j . Here we may utilize simple univariate Chebyshev interpolation at fixed values of s , see [35]. Errors in this interpolation converge spectrally fast in the number of Chebyshev modes M .

3. Periodic compatibility conditions

For both the Poisson and Stokes problems (and potentially other PDE), it will be necessary to slightly modify the simple algorithm presented in Section II due to the constant null-space in the periodic operators used to solve the regular problem. We discuss the Poisson case here; the adjustments required for the Stokes problem are no different. In particular, we have the solvability condition that $\Delta u = \eta f$ is solvable on a periodic domain only if ηf has mean 0; that is, if $\int_{\mathcal{C}} \eta f = 0$. This will typically not be the case. There are several solutions to this problem; here, we use perhaps the simplest — modifying the function ηf *only in the exterior of Ω* . This is accomplished by computing the integral of ηf over Ω , and adding a regularized *bump* function in the domain with a countervailing mass. To be precise, we let $\xi(\mathbf{x})$ be a bump function with $\int_{\mathcal{C}} \xi = 1$ and $\text{supp } \xi$ disjoint from Ω . We may then solve:

$$\Delta u = \eta f - \xi \int_{\mathcal{C}} \eta f, \quad (10)$$

where the right-hand side clearly satisfies the solvability condition and is unchanged within $\tilde{\Omega}$ (actually, Ω). The numerical approximation may be affected, if, however, ξ is poorly resolved by the discretization. One solution is to use the bump function from which η is constructed, see Section III A 4. However, because η is constructed as an integral of the bump function, it is better resolved than the underlying bump. Instead, we use that bump function but spread out over twice the radius, and have found empirically that this does not affect the overall error in the method.

A final question is how to place the center of the bump function ξ . In many cases, this can be placed by the user; for example, in the problem shown in Figure 2, the bump could be manually centered in the upper right hand corner, and

the computational domain \mathcal{C} could be taken tight to Ω . A solution that always works, without manual intervention from the user, is to simply increase the size of the domain by $w = 2Mh$ (see Section III A 3), in which case the center can be placed at $(X_{\max} + Mh, Y_{\max} + Mh)$ and guaranteed not to overlap Ω . This simplified approach, which comes at the cost of slightly increasing the number of degrees of freedom in \mathcal{C} , is what we will use throughout this manuscript.

4. Evaluation of layer potentials

In two places — both in the stitching problem and in the homogeneous correction, we will be required to evaluate layer potentials to a large number of points, many of which may be close to the curve from which these layer potentials emanate. This presents two problems:

1. Naive evaluation near boundaries typically have $\mathcal{O}(1)$ errors that do not disappear on grid refinement [36, 37];
2. Direct evaluation is $\mathcal{O}(N_b N_t)$, with N_b and N_t giving the number of boundary points and target points, respectively. Here, $N_b = N$ and $N_t \propto N^2$, and so direct evaluation is $\mathcal{O}(N^3)$.

For all problems in this paper, we will use a recently developed method called *Quadrature by Fundamental Solutions* [38], combined with a standard Fast-Multipole method library [13], which provides a reasonable solution to both problems. This method, once setup is done, allows for highly-accurate evaluation of layer-potentials up to and on the boundary of the domain, in a kernel independent manner, in $\mathcal{O}(N_b^2 + N_t) = \mathcal{O}(N^2)$ time. Unfortunately, it does come with a $\mathcal{O}(N_b^3)$ setup cost. When N_b is relatively small — up to 10,000 or so, modern implementations of LAPACK are surprisingly fast and this is rarely the dominant cost. For larger problems, other, more scalable methods, such as panel discretizations with kernel specific close-evaluation quadratures [37], would be more efficient and maintain an asymptotic scaling of $\mathcal{O}(N^2)$, in both setup and solution stages.

IV. METHODS — SPECIFICS

We are now ready to return to solving a PDE, and provide specific algorithms for all stages of the computation. This section goes deeper into the method as developed in Section II, utilizing the tools developed in Section III. We restate our model Poisson problem here:

$$\Delta u = f \quad \text{in } \Omega, \quad (11a)$$

$$u = g \quad \text{on } \Gamma. \quad (11b)$$

For simplicity, we assume that f and g are given to the user as evaluable functions on their respective domains.

Our first step is to discretize the problem: in this section we assume that the number of boundary nodes N , the number of Chebyshev modes M , and a grid-spacing h are given. The boundary and annulus are discretized as described in Section III A 2, with $R = Mh$. If $R > R_{\max}$, no solution is attempted. A more principled way to set these parameters is given in Section VI, but we proceed in this way for now to demonstrate certain basic features that would be otherwise hidden. All discrete nodes in \mathcal{C} are categorized into physical points ($\in \Omega$) and exterior points ($\in \Omega^C$), and those that are physical points are further categorized into points within the annular region ($\in \mathcal{A}$) and those within the faithful region ($\in \tilde{\Omega}$), via the technique described in Section III B 1. The cutoff function η is then evaluated for all discrete nodes of \mathcal{C} . Since the Poisson problem has a nullspace, a regularized bump function ξ is also computed at all discrete nodes of \mathcal{C} (see Section III B 3). At this point, our domain is discretized and we have all relevant information required to solve PDEs on the domain. We evaluate f at the discrete nodes of \mathcal{C} and \mathcal{A} , and g at the discrete nodes \mathbf{X}_j .

A. Function intension, and solving the regular problem

We can now describe explicitly how to solve the *regular problem* outlined in Section II A. First, the intended function ηf is computed at all discrete nodes of \mathcal{C} by defining it to be 0 for $\mathbf{x} \in \Omega^C$ and to be ηf for $\mathbf{x} \in \Omega$. A modified f with mean 0 is computed as $f^m = \eta f - \xi \int_{\mathcal{C}} \eta f$, and its Fourier transform $\widehat{f^m}_{\mathbf{k}}$ is computed via the standard two-dimensional FFT. The Fourier modes of $\widehat{u}_{\mathbf{k}}$ are then computed as $\widehat{u}_{\mathbf{k}} = \widehat{f^m}_{\mathbf{k}} / |\mathbf{k}|^2$, with $\widehat{u}_{\mathbf{0}} = 0$, with \mathbf{k} the wavevectors for the domain \mathcal{C} . The regular solution \tilde{u} is then recovered by the standard two-dimensional inverse FFT. Since N_x and N_y are proportional to N , the total cost for solving the regular problem scales as $\mathcal{O}(N^2 \log N)$.

B. The annular problem

We now seek to solve the annular problem:

$$\Delta u_{\mathcal{A}} = f \quad \text{in } \mathcal{A}, \quad (12a)$$

$$u_{\mathcal{A}} = 0 \quad \text{on } \Gamma \text{ and on } \mathcal{I}. \quad (12b)$$

In the coordinates s and r for the annular region \mathcal{A} , the Laplace operator is given by:

$$\Delta u_{\mathcal{A}} = \frac{1}{\psi} \left[\frac{\partial}{\partial r} \left(\psi \frac{\partial u_{\mathcal{A}}}{\partial r} \right) + \frac{\partial}{\partial s} \left(\frac{1}{\psi} \frac{\partial u_{\mathcal{A}}}{\partial s} \right) \right], \quad (13)$$

where $\psi(s, r)$ is given by $\psi(s, r) = \phi(s)(1 + r\kappa(s))$, with κ the curvature of Γ . Inverting this operator is not completely trivial. For discretizations where NM is relatively small, it is probably reasonable to form Δ and directly invert it using dense linear algebra. This, however, comes at a setup cost of $\mathcal{O}((NM)^3)$ (to factor the operator), and an application cost of $\mathcal{O}((NM)^2)$. Especially for larger M , this would significantly dominate the entire computation. Instead, we seek an iterative solution to the problem.

We first note that if Γ is a circle with a uniform parametrization, then both κ and ϕ are independent of s . This means that we can take the Fourier transform to obtain:

$$\widehat{\Delta u_{\mathcal{A}k}} = \frac{1}{\psi} \left[\frac{\partial}{\partial r} \left(\psi \frac{\partial \widehat{u_{\mathcal{A}k}}}{\partial r} \right) - \frac{1}{\psi} k^2 \widehat{u_{\mathcal{A}k}} \right], \quad (14)$$

This operator can, upon discretizing $\partial/\partial r$ via standard Chebyshev operators, be formed and directly inverted, independently for each k , at a total cost of $\mathcal{O}(NM^3)$. We may thus solve the annular problem on a simple circle via the following algorithm:

1. Given f , compute \hat{f}_k via the 1D FFT. Cost: $\mathcal{O}(MN \log N)$.
2. For each mode k , invert Equation (14) to obtain $\widehat{u_{\mathcal{A}k}}$. Total cost: $\mathcal{O}(NM^2)$.
3. Compute $u_{\mathcal{A}}$ via the 1D inverse FFT. Cost: $\mathcal{O}(MN \log N)$.

While this procedure does not work for general domains, it provides a surprisingly effective preconditioner, and it is straightforward to apply the annular Laplacian given in Equation (13) for a total cost of $\mathcal{O}(MN \log N) + \mathcal{O}(NM^2)$. We have found, and will show in the examples, that GMRES with the circular preconditioner converges robustly, with a relative residual of 10^{-14} typically reached in 10-20 iterations, so long as R does not approach too close to R_{\max} . When R approaches R_{\max} , the iteration count can increase dramatically.

Remark 4 (Spectral vs. Pseudospectral). *We have implemented this solver discretizing both the nodal values and the spectral modes. For scalar problems, both methods work about equally well, although the spectral method sometimes saturates at slightly lower errors (and for this reason we use it throughout this manuscript). For vector problems (and in particular Stokes), we have found it simpler to obtain robust convergence by discretizing spectral modes, omitting from the solution vector the Nyquist frequency in the azimuthal direction, see Section XI. In both implementations, we use a rectangular method for discretizing the Chebyshev operators and imposing the boundary conditions [39].*

Remark 5 (Choice of boundary conditions). *For simplicity, we have chosen $u = 0$ as the boundary conditions for the annular solve at both \mathcal{I} and Γ . In certain circumstances, it could be beneficial to choose other boundary conditions, subject to the constraint that those conditions satisfy any compatibility conditions for the PDE. One such circumstance is if the user has a method to perform close-evaluation of either a single-layer or double-layer potential, but not both. Consider the case where the user only has a method to apply the single-layer potential. Rather than setting the boundary condition on \mathcal{I} to be 0, we could instead set it to be $\tilde{u}|_{\mathcal{I}}$. Looking briefly ahead to Equation (15), we see that then $\gamma = 0$, and so the double-layer potential in Equation (16) can be ignored. Alternatively, if the user only has a method to apply the double-layer potential, they could fix $\partial_{\mathbf{n}} u_{\mathcal{A}} = \partial_{\mathbf{n}} \tilde{u}$ at \mathcal{I} , instead.*

C. The stitching problem

We now know $\tilde{u}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{C}$, and the annular solution $u_{\mathcal{A}}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{A}$. We seek now to compute the jumps in the solution and its normal derivative:

$$\gamma = \lim_{\mathbf{x} \rightarrow \mathcal{I}^{\Omega}} \tilde{u} - \lim_{\mathbf{x} \rightarrow \mathcal{I}^{\mathcal{A}}} u_{\mathcal{A}}, \quad (15a)$$

$$\sigma = \lim_{\mathbf{x} \rightarrow \mathcal{I}^{\Omega}} \partial_{\mathbf{n}} \tilde{u} - \lim_{\mathbf{x} \rightarrow \mathcal{I}^{\mathcal{A}}} \partial_{\mathbf{n}} u_{\mathcal{A}}, \quad (15b)$$

with $\partial_{\mathbf{n}}f$ denoting the normal derivative of f and $\mathbf{x} \rightarrow \mathcal{I}^W$ denoting the limit as \mathbf{x} tends to \mathcal{I} from within the domain W . Computing \tilde{u} and $u_{\mathcal{A}}$ on \mathcal{I} can be done directly using the interpolation scheme given in Section III B 2. The normal derivatives can be computed by computing $\partial_x \tilde{u}$ and $\partial_y \tilde{u}$ via FFT based differentiation, and $\partial_r u_{\mathcal{A}}$ via Chebyshev differentiation, before again applying the interpolation operators given in Section III B 2. This immediately gives $\partial_{\mathbf{n}} u_{\mathcal{A}}$, and $\partial_{\mathbf{n}} \tilde{u}$ can be computed then as $\partial_{\mathbf{n}} \tilde{u} = \mathbf{n} \cdot \nabla \tilde{u}$.

For the Poisson problem, we may correct these jumps by adding the single and double layer potentials $\mathcal{S}_{\mathcal{I}}\sigma - \mathcal{D}_{\mathcal{I}}\gamma$, given explicitly by:

$$(\mathcal{S}_{\mathcal{I}}\sigma - \mathcal{D}_{\mathcal{I}}\gamma)(\mathbf{x}) = \int_{\mathcal{I}} G(\mathbf{x}, \mathbf{y})\sigma(\mathbf{y}) d\mathbf{s}_{\mathbf{y}} - \int_{\mathcal{I}} \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}^{\mathbf{y}}} \gamma(\mathbf{y}) d\mathbf{s}_{\mathbf{y}}, \quad (16)$$

where $G(\mathbf{x}, \mathbf{y}) = (2\pi)^{-1} \log |\mathbf{x} - \mathbf{y}|$. As discussed in Section III B 4, we utilize FMM accelerated QFS to evaluate these layer potentials, but briefly describe this here in the context of the stitching step for an interior problem. Points in $\tilde{\Omega}$ are inside \mathcal{I} , while points in \mathcal{A} are outside \mathcal{I} . These require different treatment. To evaluate to all $\mathbf{x} \in \tilde{\Omega}$, we use QFS to construct an effective inward³ representation; that is, we compute a source curve \mathcal{I}_{in} and an effective potential ζ_{in} such that:

$$(\mathcal{S}_{\mathcal{I}}\sigma - \mathcal{D}_{\mathcal{I}}\gamma)|_{\tilde{\Omega}}(\mathbf{x}) = \mathcal{S}_{\mathcal{I}_{\text{in}}}\zeta_{\text{in}}(\mathbf{x}), \quad (17)$$

for all $\mathbf{x} \in \mathcal{I}$. Note that the layer potentials on the left-hand side are singular or principal-value, and so we specify here that the interior limit (from $\tilde{\Omega}$) is taken. Once ζ_{in} has been determined, it can be evaluated from \mathcal{I}_{in} using a standard periodic trapezoid rule and FMM acceleration to all points $\mathbf{x} \in \tilde{\Omega}$:

$$(\mathcal{S}_{\mathcal{I}_{\text{in}}}\zeta_{\text{in}})(\mathbf{x}) \approx \frac{-1}{2\pi} \sum_{j=1}^{N_s} \log |\mathbf{x} - \mathbf{X}^{\mathcal{I}_{\text{in}}}(s_j)| \zeta_{\text{in}}(s_j) w_j, \quad (18)$$

where N_s is the number of points discretizing the source curve \mathcal{I}_{in} and $w_j = 2\pi\phi_j/N_s$, with ϕ_j the speed for the parametrization of \mathcal{I}_{in} . The number of source points is typically $N_s = N$, although this may need to be adjusted due to geometric constraints, see [38]; and this method for evaluating the integral Equation (16) converges spectrally fast in N . The process of evaluating the layer potential in Equation (16) at $\mathbf{x} \in \mathcal{A}$ is similar; an effective outward potential ζ_{out} on a source curve \mathcal{I}_{out} is computed, subject to the constraint that Equation (17) holds but now with the exterior limit (from \mathcal{A}) taken on the right hand side. The layer potential in Equation (18), with interior and exterior quantities swapped, is evaluated via FMM at all $\mathbf{x} \in \mathcal{A}$. We then define the *inhomogeneous solution*:

$$u_I(\mathbf{x}) = \begin{cases} \tilde{u}(\mathbf{x}) + (\mathcal{S}_{\mathcal{I}}\sigma - \mathcal{D}_{\mathcal{I}}\gamma)(\mathbf{x}) & \text{for } \mathbf{x} \in \tilde{\Omega}, \\ u_{\mathcal{A}}(\mathbf{x}) + (\mathcal{S}_{\mathcal{I}}\sigma - \mathcal{D}_{\mathcal{I}}\gamma)(\mathbf{x}) & \text{for } \mathbf{x} \in \mathcal{A}. \end{cases} \quad (19)$$

We now have a smooth particular solution u_I to the PDE in Ω . The algorithm for the stitching step is summarized in Algorithm 2, with computational scaling given for our implementation; the total effort is $\mathcal{O}(N^2 \log N) + \mathcal{O}(NM^2)$.

Algorithm 2 Stitching problem

 $\mathcal{O}(N^2 \log N + NM^2)$

Require: The Fourier transform of the regular solution $\hat{\tilde{u}}$, and the annular solution $u_{\mathcal{A}}$.

- 1: [$\mathcal{O}(N^2)$] compute $\widehat{\partial_x \tilde{u}}$ and $\widehat{\partial_y \tilde{u}}$
- 2: [$\mathcal{O}(N^2 \log N)$] interpolate \tilde{u} , $\partial_x \tilde{u}$, and $\partial_y \tilde{u}$ onto \mathcal{I} ▷ see Section III B 2
- 3: [$\mathcal{O}(N)$] compute $\partial_{\mathbf{n}} \tilde{u} = \mathbf{n} \cdot \nabla \tilde{u}$
- 4: [$\mathcal{O}(NM^2)$] compute $u_{\mathcal{A}}$ and $\partial_{\mathbf{n}} u_{\mathcal{A}}$ on \mathcal{I} ▷ see Section III B 2
- 5: [$\mathcal{O}(N)$] compute γ and σ ▷ see Equation (15)
- 6: [$\mathcal{O}(N^2)$] compute inward effective potential ζ_{in} defined on effective source curve \mathcal{I}_{in}
- 7: [$\mathcal{O}(N^2)$] Evaluate $\mathcal{S}_{\mathcal{I}_{\text{in}}}\zeta_{\text{in}}$ at all points $\mathbf{x} \in \tilde{\Omega}$ and add to \tilde{u} to obtain u_I for $\mathbf{x} \in \tilde{\Omega}$
- 8: [$\mathcal{O}(N^2)$] compute outward effective potential ζ_{out} defined on effective source curve \mathcal{I}_{out}
- 9: [$\mathcal{O}(N^2)$] Evaluate $\mathcal{S}_{\mathcal{I}_{\text{out}}}\zeta_{\text{out}}$ at all points $\mathbf{x} \in \mathcal{A}$ and add to $u_{\mathcal{A}}$ to obtain u_I for $\mathbf{x} \in \mathcal{A}$

▷ Steps 6-9, see Section III B 4

³ Note that we use the term “inward” here, rather than “interior”. Inward will be used for potentials evaluated from \mathcal{I} into $\tilde{\Omega}$, while “outward” will be used for potentials evaluated from \mathcal{I} into \mathcal{A} . For interior problems, inward corresponds to interior and outward cooresponds to exterior; for exterior problems, inward corresponds to exterior and outward corresponds to interior.

D. The homogeneous problem

Finally, we must correct u_I to satisfy the physical boundary conditions. First, u_I is interpolated to the boundary nodes \mathbf{X}_j from the annular nodes using the method described in Section III B 2. We now know the discrepancy $g - u_I$ at all discrete boundary nodes. It thus remains to solve the homogeneous equation:

$$\Delta u_H = 0, \quad \text{in } \Omega, \quad (20a)$$

$$u_H = g - u_I \quad \text{on } \Gamma. \quad (20b)$$

For this interior Dirichlet problem, the simple representation $u_H = \mathcal{D}_\Gamma \zeta$ suffices to provide the well-conditioned second-kind boundary integral equation [4]:

$$(\mathbb{I}/2 - D_{\Gamma,\Gamma}) \zeta = g - u_I, \quad (21)$$

with $D_{\Gamma,\Gamma}$ denoting the principal value operator obtained by evaluating \mathcal{D}_Γ on Γ . The left-hand side operator can be discretized with spectral accuracy in N utilizing singular Kress quadrature [40], to give the $N \times N$ matrix equation:

$$A^{ij} \zeta_j = (g - u_I)|_{s_j}, \quad (22)$$

which requires $\mathcal{O}(N^2)$ effort to form. This matrix can then be directly factored with $\mathcal{O}(N^3)$ effort, to allow its solution in $\mathcal{O}(N^2)$ time, or solved via GMRES in $\mathcal{O}(N^2)$ time via direct application, with an iteration count independent of N .

Remark 6 (FMM and Fast direct solvers). *In cases where N is large, further accelerations are possible. If the physical problem is well conditioned, using GMRES and applying A with an FMM and local corrections reduces the cost to $\mathcal{O}(N)$ []. When the physical problem itself is poorly conditioned (such as high-frequency Helmholtz problems), the number of GMRES iterations may become large. Using a fast-direct solver is one option in this case [41], which we have not yet explored.*

Finally, once ζ is known, $u_H = \mathcal{D}_\Gamma \zeta$ can be evaluated for all \mathbf{x} in both $\tilde{\Omega}$ and \mathcal{A} , again using the method described in Section III B 4. The total numerical cost of the homogeneous correction is $\mathcal{O}(N^2)$.

E. Finishing up

At this point, we know u_I and u_H for all discrete nodes \mathbf{x} of \mathcal{C} in $\tilde{\Omega}$ and discrete nodes \mathbf{x} of \mathcal{A} . We can simply add these together to obtain u . We do not, however, at this point know u on the discrete nodes \mathbf{x} of \mathcal{C} within the region \mathcal{A} . These values can now be obtained by interpolating u from \mathcal{A} to any $\mathbf{x} \in \mathcal{A}$ using the method described in Section III B 2. This gives us a full representation of u ; we can now integrate, differentiate, or interpolate u to any location in Ω with spectral accuracy (in N and M).

F. Parameters

Finally, we collect together the various parameters that need to be set in order to fully define the method. These are given in Table I. Note that the number of modes discretizing \mathcal{A} is taken to be N , and the tolerance used for *all* iterative and approximate methods is set to the same value of $\epsilon = 10^{-14}$ throughout. Because this tolerance is used in multiple approximate methods throughout the solver, we do not expect solutions to achieve this tolerance, but rather a small multiple of it.

V. A SIMPLE EXAMPLE AND DEPENDENCE ON M

We now return to the example problem shown in Figure 2, and analyze the convergence properties of the scheme, deferring a discussion of parameter selection and numerical performance to Section VI, as immediately jumping to optimal parameter choices obscures some details. The problem we will analyze is the Dirichlet Poisson problem, with f and g manufactured from the known solution $u = e^{\sin(x)} \sin(2y) + \log(0.1 + \cos(y)^2)$, set on a star-shaped domain defined by the function

$$\mathbf{X}(s) = (x_c + r\omega(s) \cos(s), y_c + r\phi(s) \sin(s)), \quad (23)$$

Parameter	Description
N	Number of discrete boundary modes
M	Number of Chebyshev modes discretizing \mathcal{A}
R	Annular radius
h	gridspacing of regular discretization of \mathcal{C}
b	bandwidth of η
ϵ	tolerance used for all iterative and approximate methods

TABLE I: Required parameter choices.

with $\omega(s) = 1 + a \cos(ds)$. For this problem we take $x_c = y_c = 0$, $r = 1$, $d = 5$, and $a = 0.15$. This is a relatively simple domain to allow more exploratory range in some of the numerical experiments done in this section; more complex domains are considered in later examples.

A. Behavior for fixed M

We begin with a simple exploration of the properties of this solver for a fixed number of Chebyshev modes M and a variable radius $R = Mh_{\min}$, with h_{\min} the smallest discrete boundary gridspacing, as given in Section III A 1. We vary N from 100 to 1500, tracking both the error, in $L^\infty(\Omega)$, and the number of GMRES iterations required to invert the annular problem, for $M = 4, 8, 12$, and 16. The regular gridspacing h is set to be $h_{\min}/2$, $b = \lceil 1.5M \rceil$, and $\epsilon = 10^{-14}$. The error and number of iterations are shown in panels (a) and (b) of Figure 5, respectively. There are several points worth making about this simple study.

1. Solutions fail to exist (or have large errors) for small N when M is large. This is because the $R < R_{\max}$ criteria is not obeyed by the configuration: the implied coordinates would be singular. When the criteria is just *barely* obeyed, a large number of GMRES iterations are required to invert the annular problem. The iteration count decays rapidly as N is increased, to a nearly M independent number of ≈ 10 .
2. Higher M indeed leads to faster convergence — the accompanying dashed lines are M th order convergence lines.
3. Rapid convergence *stagnates* at a certain error: for $M = 4$ this is off the graph, but for larger M this effect is clearly apparent. This stagnation occurs when the dominant error in the problem is the resolution of the cutoff function η^4 . Empirically, we observe continued second-order convergence once this floor has been reached⁵.
4. The horizontal gray dotted line is placed at $\epsilon = 10^{-14}$. Convergence to a small multiple of ϵ is observed for the $M = 16$ convergence curve.

We do not belabor the first or the third point because they both have the same solution: letting the number of Chebyshev modes M increase proportionally with N . In doing so, the physical width of \mathcal{A} can be fixed geometrically so that it embeds without coordinate collapse (see Section III A 1). Convergence of the annular solution is now expected to be spectral in N , and because R remains fixed but N_x and N_y increase, the regularized step function η becomes progressively better resolved, eliminating the stagnation observed when fixed values of M are used.

B. Behavior for proportional M

We now redo this refinement study, with the same parameter choices as in Section V A but now scaling M as a function of N . To be precise, we will take $M = \lceil \gamma N / 100 \rceil$, for $\gamma = 1, 2, 3$, and 4, restricting M to a minimal value of 4 and a maximal value of 40. Figure 6 shows the $L^\infty(\Omega)$ error along with the number of GMRES iterations used in the annular solve. Notice that the x -axis is now linear; as expected, convergence is spectral in N . Using smaller values of γ leads to lower iteration counts but a slower rate of exponential convergence; higher values typically lead to faster convergence, although beyond a certain point this advantage becomes negligible as other errors (or simply resolving

⁴ Note that when M is fixed, and the regular gridspacing is set proportional to the boundary gridspacing, η varies from 0 to 1 over the same number of regular gridpoints regardless of N .

⁵ The continued second order convergence once occurs because the cutoff function η multiplies f , but we are solving for u , which is two derivatives smoother.

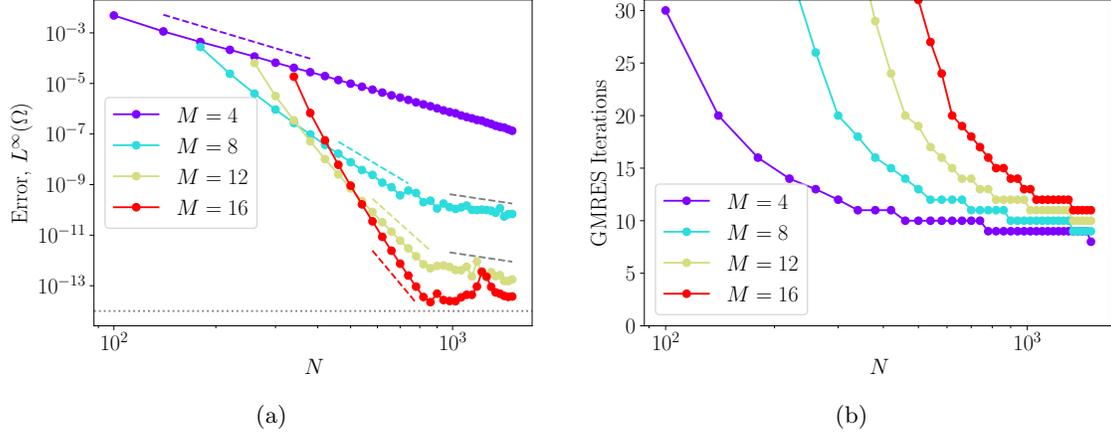


FIG. 5: Convergence study in boundary points N at fixed M . Panel (a) shows errors in $L^\infty(\Omega)$, with matching M th order convergence lines (dashed). The dashed gray lines are proportional to $1/N^2$, and the dotted gray line shows the 10^{-14} tolerance set on all approximate or iterative portions of the solver. Panel (b) shows the number of GMRES iterations used by the Annular solver to reach a relative residual $< 10^{-14}$.

f) begin to dominate. As in the $M = 16$ case, errors saturate at a small multiple of 10^{-14} . The jumps in the iteration counts and the staircase effect in the error (especially for $\gamma = 1$) are due to integer shifts in the value of M as N is increased.

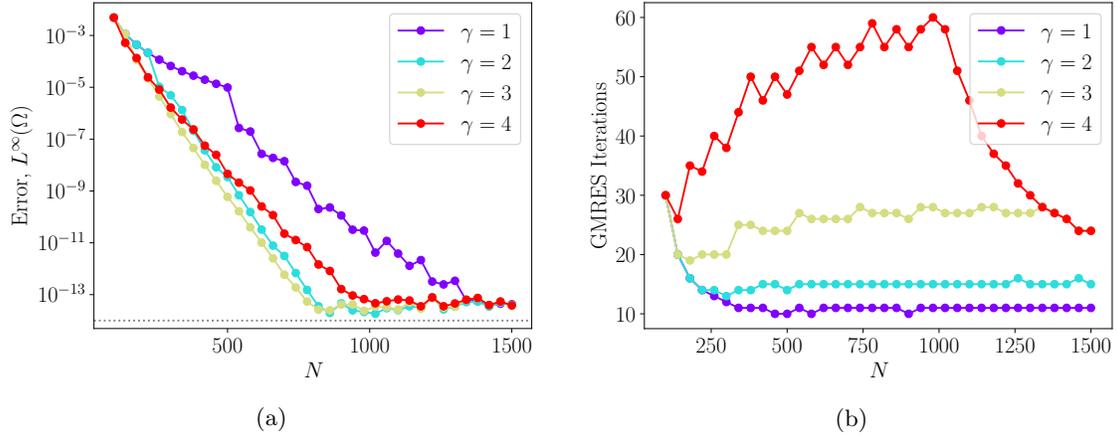


FIG. 6: Convergence study in boundary points N with fixed γ (M proportional to N). Panel (a) shows errors in $L^\infty(\Omega)$; panel (b) shows the number of GMRES iterations used by the annular solver to reach a relative residual $< 10^{-14}$.

VI. PARAMETER SELECTION

We are now ready to discuss how to select parameters. We suppose that the user provides a parametrized curve Γ that bounds the domain Ω , along with an inhomogeneity f and boundary inhomogeneity g . For simplicity, we assume that Γ , f , and g are given as evaluable functions anywhere in their respective domains. The choices we must make are given in Table I. There are many ways these choices could be made, and in a highly optimized numerical solver they would likely be chosen to minimize the error given a runtime constraint. We will choose a simpler method — enforcing that every element of the solver is resolved to a user-specified length-scale h .

A. Selection of the annular width R

Having observed that the maximal annular width R_{\max} is purely geometric (and not a function of the discretization or parametrization), we are free to choose R before discretization is considered. The primary tradeoff here is efficiency vs. speed: the choice of R is equivalent to choosing γ in Section VB, and so the rate of exponential convergence must be traded off against an increased number of GMRES iterations. We have found that a good default is $R = R_{\max}/2$; and this will be used throughout the remainder of the paper.

B. Spatial discretization

We now turn to the spatial discretization. The simplest possibility is to allow the user to specify a lengthscale h as the basic unit of parametrization, and to ensure that all components of the solver resolve this lengthscale. In particular, we take:

1. (Selection of number of boundary points N): We remind the reader that the Jacobian of the coordinate transformation is $\mathcal{J} = \phi(s)(1 - r\kappa(s))$, where ϕ and κ are the speed and curvature, respectively. We can think of the term $(1 - r\kappa(s))$ as an amplification factor for the boundary discretization. Letting $h_{\Gamma}^j = \phi(s_j)\Delta s$, where $\Delta s = 2\pi/N$ and N is the number of boundary points, we can compute that the azimuthal grid-spacing at the interface is $h_{\mathcal{I}} = \phi(s_j)(1 - r\kappa(s_j))\Delta s$. We thus choose N to be the smallest even integer such that $\min_j(h_{\Gamma}^j, h_{\mathcal{I}}^j) < h$; with it being clear that the grid-spacing at \mathcal{I} and Γ are sufficient to bound the azimuthal spacing everywhere in \mathcal{A} .
2. (Selection of the number of Chebyshev modes M): The effective radial resolution will be $h_r = \pi R/(2M)$. We select M to be the smallest integer with $h_r < h$; i.e. $M = \lceil \frac{\pi R}{2h} \rceil$.
3. Bounds for the grid are computed as discussed in Section III A 3, and N_x and N_y are chosen to be the minimal even integers with $h_x = h_y < h$, in accordance with the considerations discussed in Sections III A 3 and III B 3.

C. Choice of the bandwidth b defining η

The final choice that remains to be made is the regularization parameter defining η . For the Poisson problem, the regularization parameter $\lceil 2R/h \rceil$ provides near-optimal results. The best choice of this parameter will, in general, be PDE dependent, and will especially differ from our choice in near-identity problems (such as high- k Helmholtz or modified Helmholtz); where the smoothing feature of the underlying elliptic operator will be less apparent. In Figure 7, we show the error as a function of the regularization parameter δ that defines η for two very different Poisson problems set on different domains over a range of underlying discretization parameters h . For both problems and all values of h , the estimate $2R/h$ provides a remarkably good estimate of the best observed value.

D. Numerical timing and stability

Finally, we rerun the problem from Section V using the parameter choices outlined in this section, over a large range of h , examining both numerical timings and stability as the number of boundary points N grows large. The error, in $L^\infty(\Omega)$, is shown in Figure 8(a) for $h = 0.002$ to $h = 0.05$ (corresponding to very fine discretizations with $N \approx 4000$ to very coarse discretizations with $N < 200$). As expected, we see rapid and exponential convergence of the error to a small multiple of $\epsilon = 10^{-14}$. As N gets larger, the error stays relatively stable, with a very slow loss of accuracy, proportional to N (shown as the dashed blue curve). This mild loss of accuracy is due to the direct computation of the gradient of u used to match interface derivatives (see Section IV C); this could be remedied by exploiting other methods to estimate the off-grid derivative.

Panels (b) and (c) of Figure 8 show walk clock timings, broken into both setup (dashed lines) and solve (solid lines) and inhomogeneous (green) vs. homogeneous corrections (purple), on two different computers. Our implementation is not optimal but wall clock times are included to demonstrate what is practically achievable with a carefully implemented but not aggressively optimized code written in Python. The timings in panel (b) are from a quad-core Macbook Pro with a single Intel(R) Core(TM) i7-8569U CPU @ 2.80GHz and 16 GB of RAM, the timings in panel (c) are from a 40-core cluster node with two Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz and 768GB of RAM. Timings are broken down into both setup portions (dependent on Γ and h , but not on f or g), and solve portions (dependent on f and g). For small problem sizes (e.g. $N \approx 200$), setup and solve are both done on the 10s of milli-second timescale. When $N = 2000$, solutions are produced in < 1 second, with timings in this implementation dominated

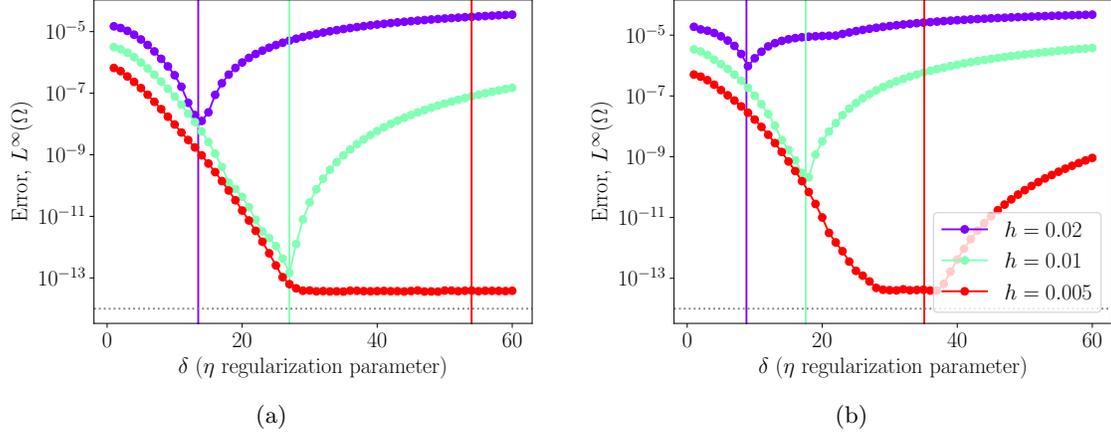


FIG. 7: Error as a function of δ (the regularization parameter for η). In both panels, different colors give different values of h , the underlying discretization scale, with purple a coarse discretization; green an intermediate discretization, and red a fine discretization. Solid vertical lines show the estimate $2R/h$ that is used to set δ in our simulations. Panel (a) show results computed on the relatively smooth function and simple domain used in Section V; Panel (b) shows results for the more complicated function and domain used in Section VIII.

by the annular solve and calls to the FMM; setup time is $< 2s$, with timing dominated by factorization of the dense homogeneous correction and QFS matrices for the close evaluation of layer-potentials. In Section XII, we discuss future implementational improvements that could reduce the cost of the method.

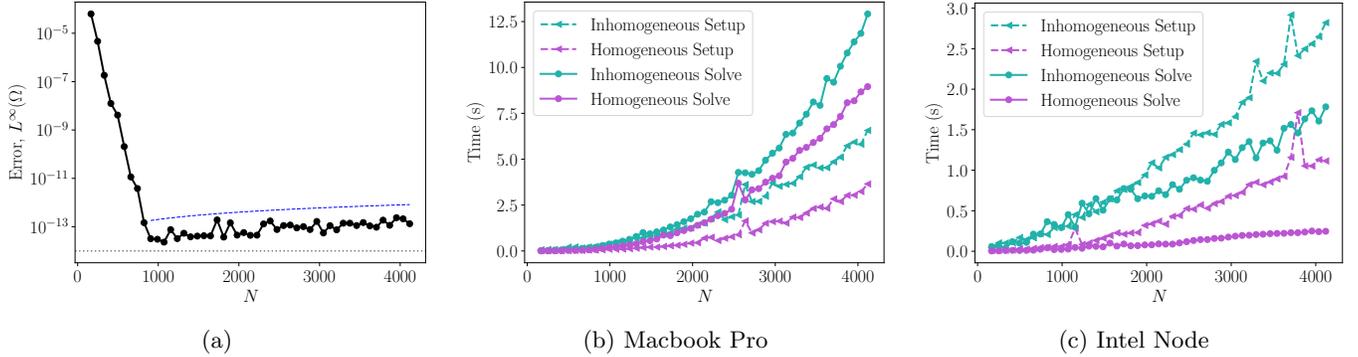


FIG. 8: Results for the simple example problem studied in Section V, using the parameter choices discussed in Section VI, over a large range of spatial discretization scales h . Panel (a) shows errors as a function of N (note that $N \propto h^{-1}$). The dotted gray line is set at $\epsilon = 10^{-14}$. The dashed blue line is proportional to N , showing the expected (slow) loss of error as N grows. Panels (b) and (c) show wall clock times on a Macbook Pro and Intel Cluster node, respectively, broken down into setup and solve times for both the inhomogeneous portion of the solve and the homogeneous correction.

VII. SUMMARY OF ALGORITHM

Finally, we present summaries of the full algorithm, with computational costs for our specific implementation given for all compute-heavy steps. We report asymptotic scalings in only N and M , taking $N_x \propto N_y \propto N$. The algorithm for the setup portion of the solve, which depends on the domain Ω , the user-specified resolution scale h , and tolerance ϵ , is given in Algorithm 3; the total scaling for this stage, in our implementation, is $\mathcal{O}(N^3) + \mathcal{O}(NM^3)$. The algorithm for the solve stage, which depends on f , g , and ϵ , is given in Algorithm 4; with a total scaling of $\mathcal{O}(N^2 \log N) + \mathcal{O}(NM^2) + \mathcal{O}(NM \log N \log M)$. Recall that N is the number of *boundary* nodes, and so the total number of unknowns is $\mathcal{O}(N^2)$, with the complexity of the FFT used to solve the regular problem setting our benchmark scaling of $\mathcal{O}(N^2 \log N)$.

Algorithm 3 Setup procedure for domain Ω $\mathcal{O}(N^3 + NM^3)$ **Require:** Smooth parametrized coordinates \mathbf{X} for the curve $\Gamma = \partial\Omega$, length-scale h , tolerance ϵ .

-
- 1: compute R_{\max} ; $R = R_{\max}/2$ ▷ see Section III A 1
 - 2: select number of boundary nodes N , Chebyshev nodes M , and regular grid nodes N_x, N_y , so all discretizations will resolve lengthscale h ▷ see Section VI B
 - 3: **if** compatibility conditions are required **then** ▷ see Section III B 3
 - 4: augment \mathcal{C} to provide space for bump function ▷ see Section III B 3
 - 5: Discretize $\mathcal{C}, \Gamma, \mathcal{A}$ ▷ see Sections III A 2 and III A 3
 - 6: [$\mathcal{O}(N^2 \log N)$] Categorize discrete nodes of \mathcal{C} and compute cutoff η ▷ see Sections III A 4 and III B 1
 - 7: **if** compatibility conditions are required **then** ▷ see Section III B 3
 - 8: compute bump function ξ ▷ see Section III B 3
 - 9: [$\mathcal{O}(N^3)$] Compute inward QFS representation for Γ ▷ see Section III B 4
 - 10: [$\mathcal{O}(N^3)$] Compute inward and outward QFS representations for \mathcal{I} . ▷ see Section III B 4
 - 11: [$\mathcal{O}(N^3)$] Compute and factor singular matrix A ▷ see Equation (22)
 - 12: [$\mathcal{O}(NM^3)$] Construct preconditioner for annular solver ▷ see Section IV B
-

Algorithm 4 Solution procedure given f, g $\mathcal{O}(N^2 \log N + NM^2 + NM \log N \log M)$ **Require:** Setup procedure as given in Algorithm 3; evaluable functions f and g **or** discrete values of (1) f known at the nodes of \mathcal{C} in $\tilde{\Omega}$ and nodes of \mathcal{A} and (2) g known at nodes of Γ ; tolerance ϵ .

-
- 1: [$\mathcal{O}(N^2)$] compute ηf at all discrete nodes of \mathcal{C} ▷ see Section IV A
 - 2: **if** compatibility conditions are required **then** ▷ see Section III B 3
 - 3: [$\mathcal{O}(N^2)$] adjust ηf to have 0 mean ▷ see Section III B 3
 - 4: [$\mathcal{O}(N^2)$] solve regular problem to get \tilde{u} at all discrete nodes of \mathcal{C} ▷ see Section IV A
 - 5: [$\mathcal{O}(NM^2)$] solve annular problem to get $u_{\mathcal{A}}$ at all discrete nodes of \mathcal{A} ▷ see Section IV B
 - 6: [$\mathcal{O}(N^2 \log N) + \mathcal{O}(NM^2)$] solve stitching problem to get u_I ▷ see Section IV C and algorithm 2
 - 7: [$\mathcal{O}(NM^2)$] Compute boundary discrepancy at Γ ▷ see Section IV D
 - 8: [$\mathcal{O}(N^2)$] Solve BIE problem and add u_H to u_I to find \mathbf{u} ▷ see Section IV D
 - 9: [$\mathcal{O}(NM \log N \log M)$] Compute \mathbf{u} at all discrete nodes of \mathcal{C} in \mathcal{A} ▷ see Section IV E
-

VIII. A MULTI-BODY PROBLEM WITH HIGH-FREQUENCY f AND COMPARISON TO PUX

In this section, we extend our method to multiply connected domains, and solve a more complex problem previously solved in [18], comparing errors with those generated by the high-order *Partition of Unity Function Extension Method* (PUX). The definition of the domain and oscillatory RHS forcing are given in [18] and are not repeated here.

A. Considerations for multiply-connected domains

Extension to multiply connected domains, with K well-separated bodies where the annular domains do not overlap, is relatively straightforward and so we lay out only the relevant considerations here.

1. Annular regions \mathcal{A}_i are defined independently for each boundary, and the choice of R_i, N_i and M_i is made independently to be consistent with the single scale h , as described in Section VI.
2. The faithful domain is defined as $\tilde{\Omega} = \Omega \setminus (\cup_i \mathcal{A}_i)$.
3. A single function η is again computed, with $\eta(\mathbf{x}) = 1$ for all $\mathbf{x} \in \tilde{\Omega}$, $\eta(\mathbf{x}) = 0 \in \Omega^C$, and transitioning in the same way over each radial region \mathcal{A} . Note that regularization parameter $\lceil 2R_i/h \rceil$ for each region will typically be different. Errors will often be controlled by the boundary with the smallest R ; this is somewhat inevitable when using global discretizations, as in this paper.
4. The annular solutions $u_{\mathcal{A}_i}$ are computed independently.
5. Solution of the homogeneous problem, and evaluation of the solution in the multi-boundary context is described in [38].

The stitching problem in the multi-body case is straightforward, but some simple optimizations exist when utilizing our specific implementation which make its computation more efficient, and we describe these here. Recall that the goal is to evaluate the layer potential given in Equation (16) for all discrete nodes of $\mathcal{C} \in \tilde{\Omega}$ and all discrete nodes of \mathcal{A}_i for each body. The first step is to compute all inward sources $\mathcal{I}_{\text{in}}^i$ along with all inward effective potentials ζ_{in}^i . Using a single FMM, these can be evaluated at all $\mathbf{x} \in \tilde{\Omega}$ and on all interfaces \mathcal{I}^i :

$$v(\mathbf{x}) = \sum_{i=1}^K \frac{-1}{2\pi} \sum_{j=1}^{N_s^i} \log |\mathbf{x} - \mathbf{X}^{\mathcal{I}_{\text{in}}^i}(s_j^i)| \zeta_{\text{in}}^i(s_j^i) w_j^i, \quad (24)$$

which requires only $\mathcal{O}(KN) + \mathcal{O}(N_{\tilde{\Omega}} + KN)$ operations (with $N_{\tilde{\Omega}}$ the number of discrete nodes of \mathcal{C} within $\tilde{\Omega}$, and N_s^i the number of source nodes on \mathcal{I}_i , with $N_s^i \propto N$ for all i). It remains to evaluate these potentials at each annulus \mathcal{A}_i . We first independently compute outward sources $\mathcal{I}_{\text{out}}^i$ and effective potentials ζ_{out}^i for each body, and on each interface we now compute an adjusted v :

$$v_{\text{adj}}^i(\mathbf{x}) = v(\mathbf{x}) + \frac{1}{2\pi} \sum_{j=1}^{N_s^i} \log |\mathbf{x} - \mathbf{X}^{\mathcal{I}_{\text{in}}^i}(s_j^i)| \zeta_{\text{in}}^i(s_j^i) w_j^i, \quad (25)$$

which is the layer potential at \mathcal{I}_i generated by all inward sources other than the i th source itself. Again, using the methodology described in [38], we compute a second outward potential ξ^i for each body so that:

$$\mathcal{S}_{\mathcal{I}_{\text{out}}^i} \xi^i = v_{\text{adj}}^i, \quad (26)$$

for all $\mathbf{x} \in \mathcal{I}^i$. We may now, independently for each body, compute:

$$\nu^i(\mathbf{x}) = \sum_{i=1}^K (\mathcal{S}_{\mathcal{I}^i} \sigma^i - \mathcal{D}_{\mathcal{I}^i} \gamma^i) |_{\mathcal{A}^i}(\mathbf{x}) = \sum_{j=1}^{N_s^i} \log |\mathbf{x} - \mathbf{X}^{\mathcal{I}_{\text{out}}^i}(s_j^i)| [\zeta_{\text{out}}^i(s_j^i) + \xi^i(s_j^i)] w_j^i, \quad (27)$$

valid for any $\mathbf{x} \in \mathcal{A}^i$. Note that this last step requires K independent FMM calls, for a total cost of $\mathcal{O}(KNM)$; these calls are embarrassingly parallel. Finally, we define the particular solution u_I in analogy to Equation (28), as:

$$u_I(\mathbf{x}) = \begin{cases} \tilde{u}(\mathbf{x}) + v(\mathbf{x}) & \text{for } \mathbf{x} \in \tilde{\Omega}, \\ \mathbf{u}_{\mathcal{A}_i}(\mathbf{x}) + \nu^i(\mathbf{x}) & \text{for } \mathbf{x} \in \mathcal{A}_i. \end{cases} \quad (28)$$

B. Comparison to PUX

We solve this problem over a range of h , although results are reported with respect to the number of gridpoints discretizing \mathcal{C} in the x -direction (N_x), for consistency with [18]. Results from that manuscript have been extracted [42] and multiplied by $\|u\|_{L^\infty} \approx 1.35$, as we report absolute errors here. We see that both methods converge rapidly to low error; although our method typically delivers slightly better accuracy for a given N_x . We make several comments, however, regarding the direct comparability of the results.

1. The parameters and functional form of the inner boundary used in [18] do not seem to generate exactly the domain shown in their manuscript. We have instead changed the non-zero values of c and d to be $c_3 = c_6 = 0.1$, $d_{-3} = 0.05$ which generates a qualitatively similar domain (see [18] for details). This may shift the error curves to some degree.
2. The results reported in the PUX method used a fixed, and very fine, discretization for the boundary (as well as the number of partitions used for the extensions), varying only the gridspacing of the underlying grid. In our results, the boundary shares a discretization scale h consistent with the background grid, as defined in Section VI.
3. The PUX method generates solutions when N is very small (say, 100). Our method (at least with the standard way of setting parameters given in Section VI), is unable to generate solutions here as the implied M would be less than 4, the minimal M we allow.

While direct comparison is not possible, it is clear that both methods produce discretizations for this problem that converge rapidly as they are refined, and deliver comparable errors.

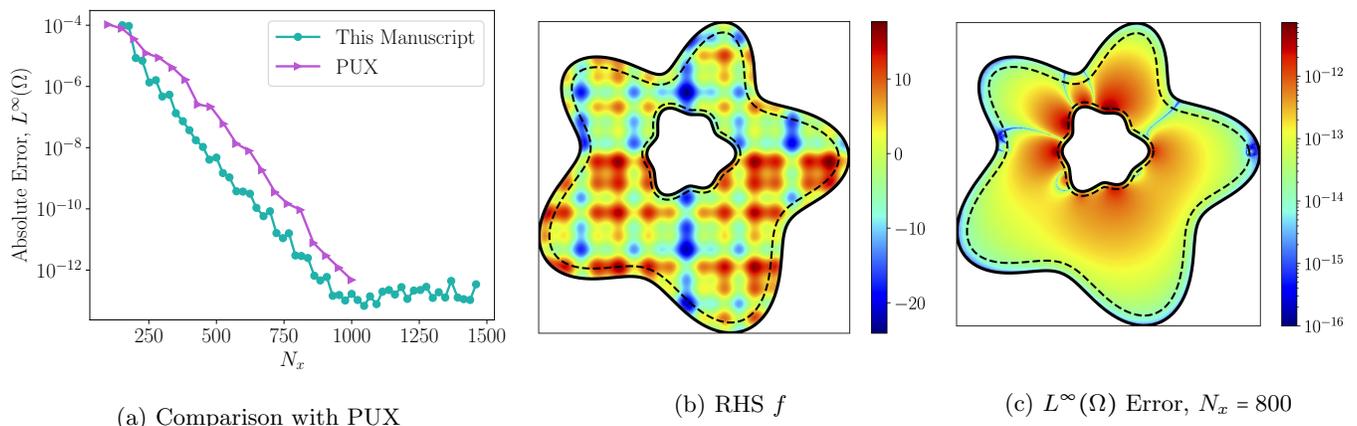


FIG. 9: Results for multi-boundary Poisson problem with high frequency RHS f from [18]. Panel (a) shows Absolute $L^\infty(\Omega)$ errors for both our method and PUX. Panel (b) shows the right-hand side f and the domain; Panel (c) shows absolute errors for $N_x = 800$.

IX. COMPARISON TO FOURIER-CONTINUATION METHOD

In this section we compare to the 2D Fourier-continuation method, as presented in [23]. In particular, we solve the Poisson problem with right-hand side $f = -\sin(kx)\sin(ky)$ set on the “kite-shaped” domain given by $\mathbf{X}(s) = (\cos(\theta) + 0.35\cos(2\theta) - 0.35)\hat{\mathbf{x}} + 0.7\sin(\theta)\hat{\mathbf{y}}$. To enable direct comparison, we solve for a smooth right-hand side with $k = 2\pi$ (see Example 4.2 and Table 3 in [23]), and a highly oscillatory right-hand side, with $k = 40\pi$ (see Example 4.3 and Table 4 in [23]), and report the most comparable error diagnostics possible.

In Figure 10, we show refinement studies for both values of k , with results from this manuscript and those reported in [23], along with a plot of the domain and pointwise error when $k = 40\pi$ and $h = 0.0025$. When $k = 2\pi$, the right-hand side is smooth. Errors (shown in panel a) from our method converge spectrally, achieving near-machine precision when $h \approx 0.005$. At large h , our method produces larger errors, with crossover achieved for h slightly less than 0.01. The reason for larger errors at large h is simple: the domain here, shown in panel (c), has high-curvature regions where it is *convex*, and low-curvature regions where it is *concave*. For an interior problem, function *intension* is hard in high-curvature convex regions: normal coordinates moving into the domain cross quickly, forcing R_{\max} to be small. The reverse is true of function *extension*, which will have more difficulty when high-curvature regions occur in concave regions for interior problems. Panel (b) shows a similar refinement study, but now with $k = 40\pi$. In this case, the right-hand side is more oscillatory, and by the time it is well resolved M is large despite the fact that R_{\max} is small; and so the higher-order convergence achieved by our scheme is sufficient to provide lower errors at all values of h . Panel (c) shows the domain, along with the associated errors when $k = 40\pi$ and $h = 0.0025$.

We use this example as a way to further analyze domains on which function *intension* vs. *extension* will have an easier time. For interior problems, *intension* requires more resolution when there are high-curvature convex regions. The reverse is true for exterior problems: high-curvature convex regions present little problem, but high-curvature concave regions are challenging. To get a handle on this, we compare the solutions generated on this domain, for both interior and exterior problems, across a range of values of k . We first show results for both an interior and exterior problem, with $h = 0.015$ and $k = 2\pi$, in Figure 11(a-b). For the exterior problem, an outer confining circle is added. As expected, for this smooth, low-frequency problem, errors are far lower — by about 6 orders of magnitude at this value of h — for the exterior problem than the interior problem. In the exterior problem, R_{\max} is large, and so M can be big — here $M = 40$. For the interior problem, M is instead 6. Errors are dominated by resolving the geometry, rather than the function.

In Figure 11(c), we show relative $L^2(\Omega)$ errors for both interior problems (solid lines) and exterior problems (dashed lines) across a range of values of k , with the most purple line corresponding to $k = 2\pi$ and the most red line corresponding to $k = 38\pi$, incrementing by 4π . The black line shows the results from the FC method for $k = 2\pi$ (reported only for the interior problem). As expected, errors produced by function *intension* on the exterior problem are smaller at all discretizations than those produced for the interior problem. To better analyze the error in these, we compute the ratio between these two errors, shown in Panel (d). These differences are most extreme at small k . In this case, achievable errors are severely limited in the interior problem by our ability to resolve the geometry, with the ratio maximized and very large (over 7 digits) at small h . As k is increased, the length-scale of the RHS becomes more comparable to, and eventually smaller than, the length-scale associated with the boundary curvature, and the

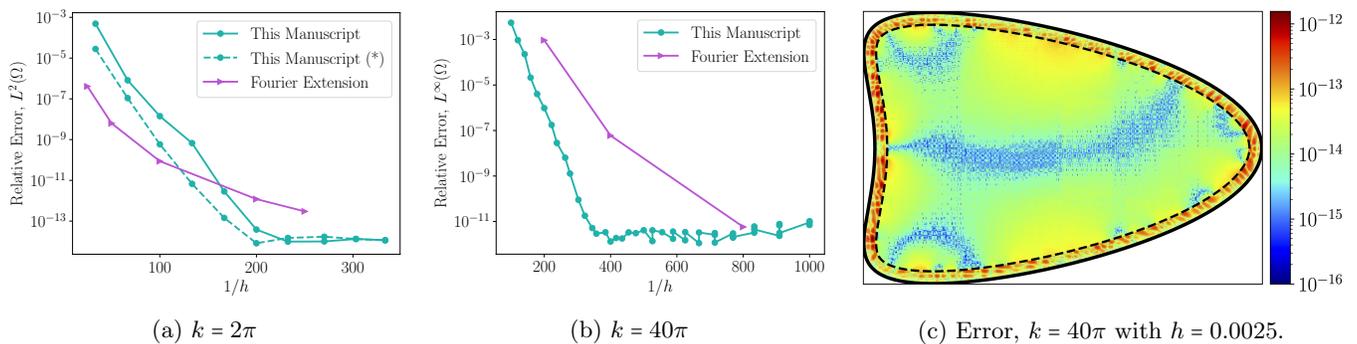


FIG. 10: Comparison to Fourier continuation method from [23]. In both problems $f = -\sin(kx)\sin(ky)$. Panel (a) shows relative $L^2(\Omega)$ errors for $k = 2\pi$. The solid green line shows our errors from our solution with parameters set as described in Section VI; the dashed line (indicated with an asterisk), shows solutions with a more aggressive radial region set by $R = 0.75R_{\max}$. Panel (b) shows relative $L^\infty(\Omega)$ errors for the harder problem with $k = 40\pi$; note as well that our solutions are $L^\infty(\Omega)$ over the *whole domain*, while those reported in [23] are reported only far from the boundary Γ . Panel (c) shows the domain for this problem, along with the error computed by our scheme when $h = 0.0025$.

ratio is reduced to ≈ 3 digits.

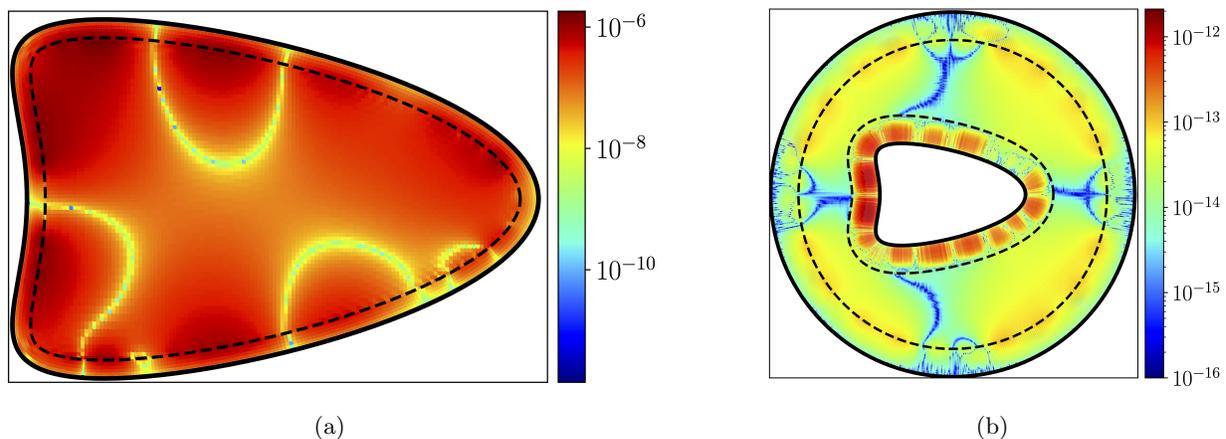


FIG. 11: Domains and errors for the interior (a) and exterior (b) problem set on the kite-shaped domain with smooth right-hand side from [23]. The kite-shaped domain here is the same physical size in each image, and $h = 0.015$ is the same. Because the domain has high curvature regions where it is convex, M is forced to be much smaller for the interior case ($M = 6$) than the exterior case ($M = 40$). When the right-hand side is much smoother than the geometry, as in this case, the difference in errors can be large.

X. THE MODIFIED-HELMHOLTZ PROBLEM

We now turn our attention to solving the inhomogeneous modified-Helmholtz problem. This problem arises when discretizing the heat equation in time: consider the simplest Forward-Euler/Backward-Euler IMEX scheme for the discretization of $u_t - \nu\Delta u = f$, which gives:

$$(\mathbb{I} - \nu\Delta t\Delta)u(t + \Delta t) = u(t) + \Delta t f(t), \quad (29)$$

subject to appropriate boundary conditions (if f is analytically known, it can be taken at $t + \Delta t$, but typically f is generated by non-linear terms, e.g. reactions or advection). Dividing through by $\nu\Delta t$ gives:

$$(\alpha^2\mathbb{I} - \Delta)u(t + \Delta t) = \alpha^2 u(t) + \alpha^2 \Delta t f(t), \quad (30)$$

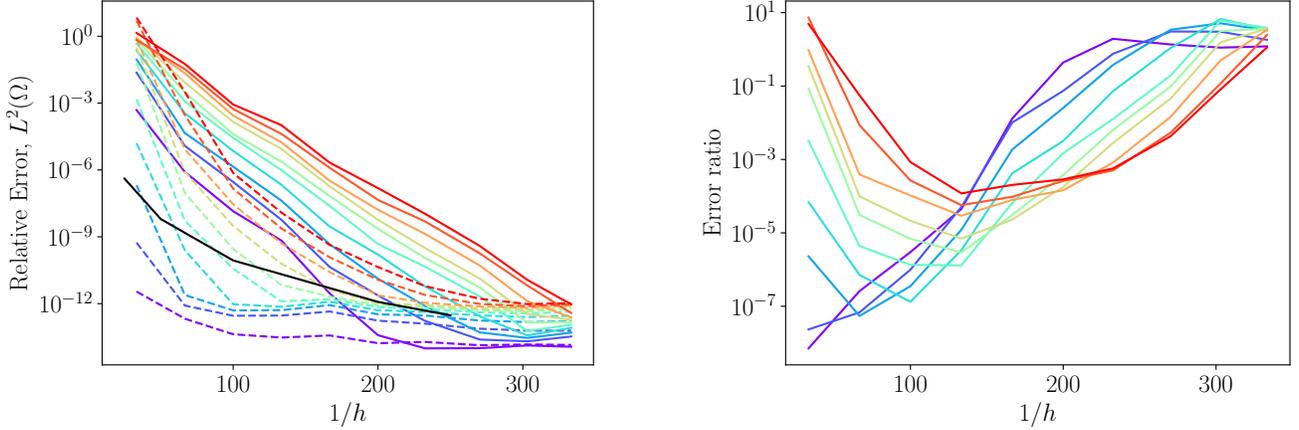


FIG. 12: Panel (a) Shows relative L^2 errors on both the interior and exterior domains (see Figure 11) over a range of h values. Errors associated with the interior domain are shown as solid lines; errors associated with the exterior domain are shown as dashed lines. The results for the same interior problem produced by the Fourier continuation method, are shown as a black line [23]. Panel (b) shows the ratio between the errors from the exterior and interior solutions (exterior error / interior error).

with $\alpha = 1/\sqrt{\nu\Delta t}$. Clearly if the diffusion coefficient ν , the timestep Δt , or both are small, α can be (very) large. We thus seek to solve the problem

$$(\alpha^2\mathbb{I} - \Delta)u = f \quad \text{in } \Omega, \quad (31a)$$

$$u = g \quad \text{on } \Gamma, \quad (31b)$$

typically for $\alpha \gg 1$. Only minor modifications need to be made to the algorithm as presented so far. We collect these here:

1. Function intension is unchanged, and the regular problem is changed only in that the Fourier symbol of the differential operator is now $\alpha^2 + \mathbf{k}^2$. This operator, unlike the Poisson operator, is invertible, with no solvability condition, and so the uniform grid can be taken tight to the boundary.
2. The method for the annular problem is nearly unchanged. Equation (14) becomes instead:

$$\mathcal{F}[(\alpha^2\mathbb{I} - \Delta)u_{\mathcal{A}}]_k = -\frac{1}{\psi} \left[\frac{\partial}{\partial r} \left(\psi \frac{\partial \widehat{u}_{\mathcal{A}k}}{\partial r} \right) + \left(\frac{1}{\psi} k^2 - \psi \alpha^2 \right) \widehat{u}_{\mathcal{A}k} \right], \quad (32)$$

which is again inverted using GMRES preconditioned by the (separable) inverse for a circular annulus.

3. The radially symmetric Green's function is now $G_\alpha(r) = \alpha^2 K_0(\alpha r)/(2\pi)$. Because $K_0(x) = -\log x + \xi(x)$ where ξ is smooth, jump conditions for the single and double layer potentials are, up to constants, the same as those for $\log r$. The stitching step is thus the same, with jumps in the value corrected by double-layer potentials and jumps in the normal derivative corrected by single-layer potentials.
4. The evaluation of these layer potentials is again done using the QFS-B algorithm from [38], with the singular on-surface evaluation done using 16th-order Alpert quadrature [40]. When α is large, if the definition of the source curve used in [38] is used, $G_\alpha(r)$ decays too rapidly to convey information between the source and check curves. To avoid this, we upsample the source curve, and move it towards the boundary, by a factor of $\max(1, h\alpha)$. Note that this upsampling factor is only large when $h \gg 1/\alpha$, in which case the problem is poorly resolved.
5. The homogeneous correction problem is unchanged, and its solution is computed by solving a well-conditioned second-kind BIE; singular integral operators are computed using 16th-order Alpert quadrature [40].

To analyze the convergence and behavior as a function of α , we solve the Dirichlet problem generated by the solution $u = \cos(20\sqrt{x^2 + y^2})$, set on a multiply connected domain similar (though not identical) to the domain on which this same problem was analyzed in [19], with solutions generated by a modified version of the PUX method, to which we compared our Poisson solver in Section VIII. The solution function u and domain are shown in Section X, along with

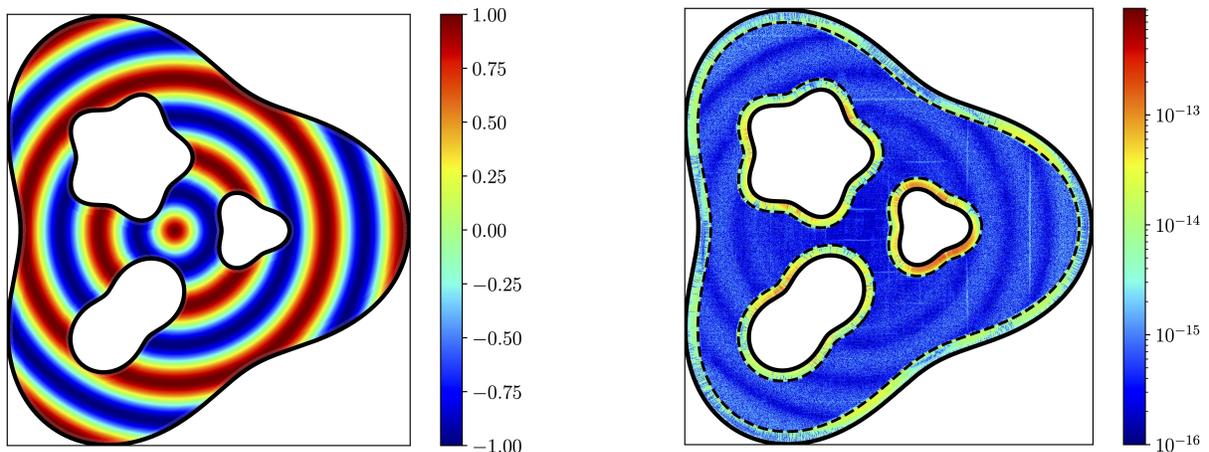


FIG. 13: Domain, solution, and errors to the modified-Helmholtz problem studied in Section X. Panel (a) shows the solution u ; panel (b) shows the error when $h = 0.002$ and $\alpha^2 = 10^5$. The $L^\infty(\Omega)$ error in this case is $< 10^{-12}$.

the pointwise error when $h = 0.002$ and $\alpha^2 = 10^5$. Even for this relatively large α , we are able to achieve a full 12 digits of accuracy, with errors in the annular solve evidently dominant.

We now solve over a range of values of $h \in [0.00125, 0.01]$ and $\alpha^2 = 10^0, 10^1, \dots, 10^6$, with results shown in Section X, although we have plotted the errors against the number of points discretizing our uniform grid in the x -direction, to enable comparison to the PUX method. For α^2 , up to 10^5 , we observe fast and stable convergence to $\approx 10^{-12}$, with the convergence curve shifted up slightly for higher values of α . For $\alpha^2 = 10^6$, our method requires markedly finer discretizations, and fails to reliably produce more than 10 digits. The slower convergence is perhaps not surprising: the length scale associated with the modified-Helmholtz equation in this case is 0.001; the finest discretization we test is $h = 0.00125$, and so it is perhaps remarkable that we achieve near-machine precision solutions when a length-scale in the problem is under-resolved; this may simply be due to the choice of a test problem whose solution does not have this length-scale present. The FMM library we were utilizing failed for higher values of α than this. Our results are qualitatively similar to those observed by the PUX method [19]. For comparison, we also plot the errors associated with their method, for $\alpha^2 = 10^1$ and $\alpha^2 = 10^5$. Because our domains are not exactly the same, we caution that comparing the errors directly is unwarranted⁶; however, both methods, perhaps surprisingly, produce nearly exactly the same rate of exponential convergence. Just as with our method, errors are slightly worse when α is large. The only minor difference between the results is that our method appears to be slightly more stable for large N at high α , finding about two more good digits in the solution when $\alpha = 10^5$.

XI. STOKES

Finally, we solve a Stokes problem with Dirichlet boundary conditions:

$$-\Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad (33a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad (33b)$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma. \quad (33c)$$

As with the modified-Helmholtz problem shown in Section X, only minor modifications to the method must be made:

1. Both components of the force \mathbf{f} must be independently adjusted to have 0 mean on \mathcal{C} (see Section IIIB3).
2. The method for the annular problem is somewhat more complicated. In our local coordinates system, the

⁶ Indeed, it is likely that PUX is producing slightly better errors for a given h , as their domains necessarily include room for an extension.

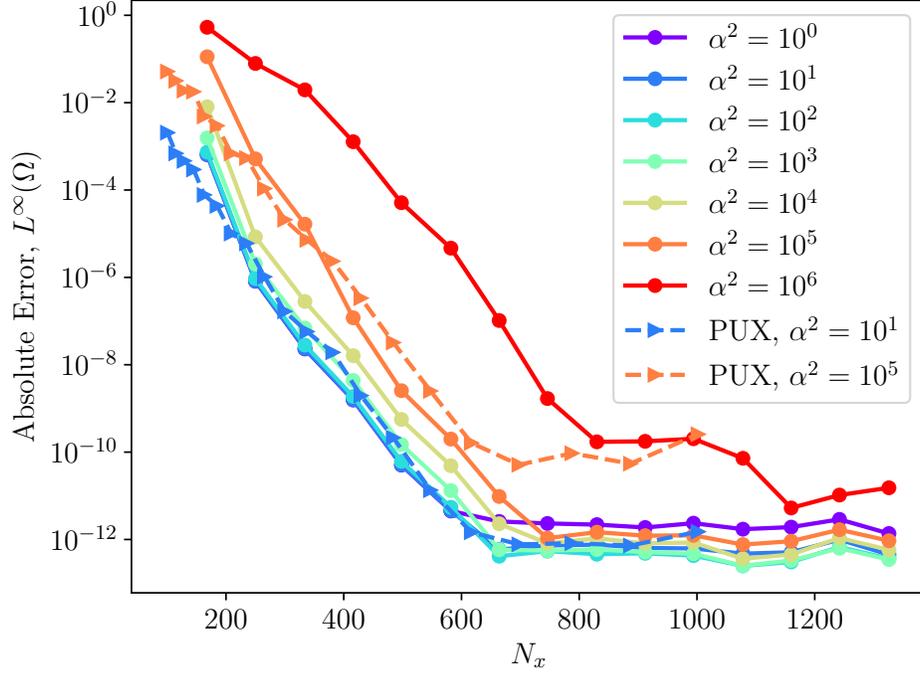


FIG. 14: Refinement study for the modified Helmholtz problem studied in Section X. Solid lines are from the method introduced in this manuscript, with the indicated value of α . Dashed lines are generated by the PUX method, extracted from Figure 8 in [19, 42], again with the indicated value of α . Both methods show exponential convergence with the same rate.

Laplacian of a vector field \mathcal{A} is given by [43]:

$$(\Delta \mathcal{A})_r = \Delta A_r - \frac{2}{\psi^2} \frac{\partial \psi}{\partial r} \frac{\partial A_t}{\partial t} - \frac{A_r}{\psi^2} \left(\frac{\partial \psi}{\partial r} \right)^2 - \frac{A_t}{\psi} \frac{\partial}{\partial r} \left(\frac{1}{\psi} \frac{\partial \psi}{\partial t} \right), \quad (34a)$$

$$(\Delta \mathcal{A})_t = \Delta A_t + \frac{2}{\psi^2} \frac{\partial \psi}{\partial r} \frac{\partial A_r}{\partial t} + \frac{A_r}{\psi} \frac{\partial}{\partial t} \left(\frac{1}{\psi} \frac{\partial \psi}{\partial r} \right) - \frac{A_t}{\psi^2} \left(\frac{\partial \psi}{\partial r} \right)^2, \quad (34b)$$

the gradient of a scalar field is:

$$\nabla p = \frac{\partial p}{\partial r} \hat{r} + \frac{1}{\psi} \frac{\partial p}{\partial t} \hat{t}, \quad (35)$$

and the divergence of a vector field is given by:

$$\nabla \cdot \mathbf{u} = \frac{1}{\psi} \left[\frac{\partial}{\partial r} (\psi u_r) + \frac{\partial u_t}{\partial t} \right], \quad (36)$$

where in all cases Δ denotes the scalar Laplacian defined in Equation (13). For the case of a circular annulus, the Stokes equations in these coordinates is separable, although the precise formulae are unwieldy, and writing them out offers no further insight. Our algorithm is the same as before: invert the Stokes operator for the real geometry, utilizing the inverse of the Stokes operator on the circular geometry as a preconditioner, with one significant caveat: discretizing all of the Fourier modes and Chebyshev modes leads to a checkerboard type instability in the pressure which prevents robust convergence of the iterative scheme. This is easily remedied by omitting the modes associated with the azimuthal Nyquist frequency.

3. The Green's function is now somewhat more complicated, and while the stitching step is the same at an abstract level, in this case jumps in the value of the velocity are corrected by double-layer potentials and jumps in the traction are corrected by single-layer potentials. Methodology for both the homogeneous correction problem and the close-evaluation of layer potentials can be found in [38].

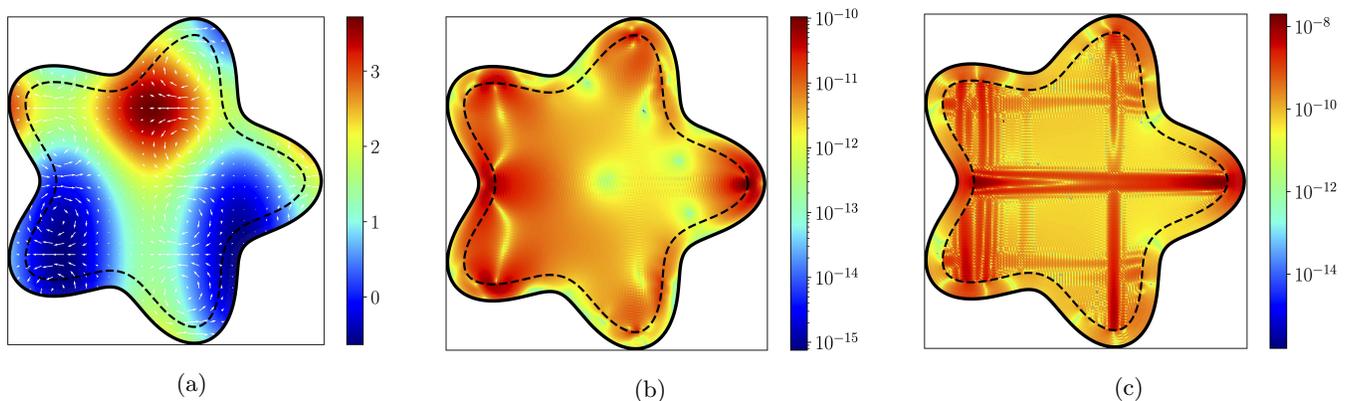


FIG. 15: Solution and errors for the Stokes problem analyzed in Section XI. Panel (a) shows pressure (pseudocolor) and velocity field (white arrows) for the analytic solution. Panels (b) and (c) show relative $L^\infty(\Omega)$ errors for \mathbf{u} and p , respectively, when $h = 0.01$.

4. The pressure is defined only up to a constant.

We now compare function intension against the 3rd-order Immersed Boundary Smooth Extension solver [17]. The solution and force are manufactured from the solution:

$$u(x, y) = e^{\sin ax} \cos by, \quad (37a)$$

$$v(x, y) = -\frac{a}{b} \cos ax e^{\sin ax} \sin by, \quad (37b)$$

$$p(x, y) = \cos cx + e^{\sin dy} \quad (37c)$$

where $\mathbf{u} = u\hat{\mathbf{x}} + v\hat{\mathbf{y}}$. For this comparison we take $(a, b, c, d) = (7, 6, 5, 3)$. The solution to this problem, with \mathbf{u} plotted as a vector field overlaying a pseudocolor plot of the pressure field, is shown in Figure 15(a), along with errors in \mathbf{u} and p in panels (b) and (c), respectively, when $h = 0.01$. We compare solutions and wall-clock timings for this problem for both function intension and the third-order IBSE method⁷, across a range of values of h , in Figure 16. Panel (a) shows relative errors for both \mathbf{u} and p . At large h , errors between the two methods are comparable, though unsurprisingly, convergence is far more rapid for function intension as h is refined, with errors for \mathbf{u} quickly reaching a small multiple of $\epsilon = 10^{-12}$. Errors in the pressure function converge at a similar exponential rate, saturating about two digits worse than \mathbf{u} . Timings, broken down into “setup” and “solve” times (with “setup” being the re-usable portion of each solve for a fixed domain Ω), are shown in Panel (b); both methods are personal implementations with similar amounts of effort expended on optimization. Across all values of h , function intension is considerably faster to setup. Once setup, IBSE produces solutions faster: solves reduce to just a few steps, dominated the FFT and LAPACK calls. Function intension is more complicated, with more computational work that cannot be directly farmed out to highly optimized external routines. As the discretization is refined the computational load becomes dominated by calls to FMM and NUFFT routines that scale well, and so while still slower, the speed of the FI solves begins to approach that of the IBSE method.

XII. DISCUSSION

We have presented a spectrally accurate solver for a relatively wide range of constant-coefficient elliptic PDE, which utilizes the stable process of function *intension* to convert a problem set on a general smooth domain to a problem set on a simple computational domain. In addition to solving a regular grid PDE with the smoothly truncated function acting as the right-hand side, we must additionally solve a PDE in an annulus localized along the boundary, and these solutions are then stitched together using techniques from boundary integral methods, which are also used to impose the physical boundary conditions. Although the analytic scaling to the implementation we use here is asymptotically optimal only after some boundary-dependent setup costs, this choice was made for convenience and can be remedied,

⁷ Results from IBSE generated by personal implementation.

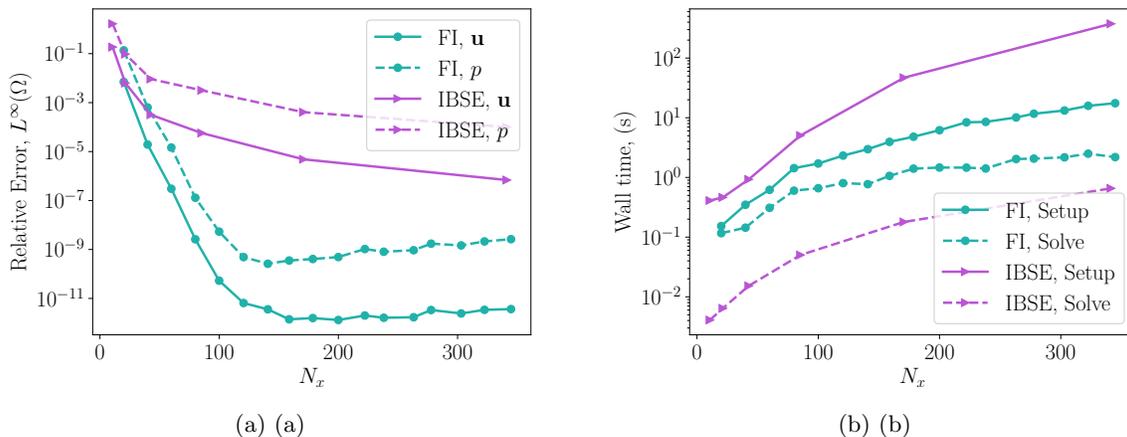


FIG. 16: Comparison between function intension and the third order IBSE method, for the Stokes problem given in Section XI. Panel (a) shows relative $L^\infty(\Omega)$ errors for \mathbf{u} and p ; panel (b) shows wall clock timings, broken down into both setup and solve portions for each respective solver.

as discussed, using established methods, reducing the asymptotic complexity to the same as the FFT used in solving the regular grid problem. Reasonably performant code implementing the method in Python is available in a repository maintained by the author [8].

To demonstrate the utility and versatility of the method, we solved Poisson, Modified-Helmholtz, and Stokes problems on a variety of domains. While we believe that the method given here has significant utility (and indeed has already found such use in [44], without a detailed presentation of the numerical method), there are cases where it suffers, the most obvious being when the problem is highly multiscale in nature. There are two separate approaches to improve applicability of the method to multiscale problems. The first is to simply replace the Fourier method used in this manuscript with an adaptive regular grid method. This fits neatly within the paradigm presented herein, necessitating only changes to the regular grid solver and interpolation operators connecting the discretization of \mathcal{C} to the annular grid \mathcal{A} , boundary Γ , and interface \mathcal{I} , although best parameter choices would need to be rethought. Full adaptivity requires further, and more fundamental changes, in particular to the discretization of Γ , the definition of the annular region \mathcal{A} , and the solver used to invert the PDE on \mathcal{A} .

XIII. ACKNOWLEDGMENTS

I owe many thanks to Mike Shelley, Shravan Veerapaneni, and Alex Barnett for a great many useful discussions, as well as Manas Rachh, for help with some of the boundary integral components and the biharmonic FMM code that underlies the Stokes solver. I'd also like to thank Dan Fortunato, for coining the rather clever term *function intension*, and for his continued work on an adaptive version of this methodology, which we hope to present soon.

-
- [1] Lloyd Nicholas Trefethen. Finite difference and spectral methods for ordinary and partial differential equations. 1996.
 - [2] Leslie Greengard and June-Yub Lee. A direct adaptive poisson solver of arbitrary order accuracy. *Journal of Computational Physics*, 125(2):415–424, 1996.
 - [3] Rainer Kress. *Linear Integral Equations*, volume 82 of *Appl. Math. Sci.* Springer, second edition, 1999.
 - [4] George Hsiao and Wolfgang L. Wendland. *Boundary Integral Equations*. Applied Mathematical Sciences, Vol. 164. Springer, 2008.
 - [5] L Greengard and M Moura. On the numerical evaluation of electrostatic fields in composite materials. *Acta Numerica*, 3:379–410, 1994.
 - [6] Wenjun Ying and J. Thomas Beale. A fast accurate boundary integral method for potentials on closely packed cells. 14:1073–1093, 2013.

⁸ For reviewers: currently located at <https://github.com/dbstein/ipde>; will be cleaned, better commented, tagged, and archived via Zenodo to accompany final manuscript.

- [7] Wen Yan, Eduardo Corona, Dhairya Malhotra, Shравan Veerapaneni, and Michael Shelley. A scalable computational platform for particulate Stokes suspensions. 416:109524, 2020. doi:<https://doi.org/10.1016/j.jcp.2020.109524>.
- [8] Bryan Quaife, Ashley Gannon, and Y-N Young. Hydrodynamics of a semipermeable vesicle under flow and confinement. *arXiv preprint arXiv:2102.00140*, 2021.
- [9] Kushal Sinha and Michael D Graham. Shape-mediated margination and demargination in flowing multicomponent suspensions of deformable capsules. *Soft matter*, 12(6):1683–1700, 2016.
- [10] Ehssan Nazockdast, Abtin Rahimian, Daniel Needleman, and Michael Shelley. Cytoplasmic flows as signatures for the mechanics of mitotic positioning. *Molecular biology of the cell*, 28(23):3261–3270, 2017.
- [11] Ehssan Nazockdast, Abtin Rahimian, Denis Zorin, and Michael Shelley. A fast platform for simulating semi-flexible fiber suspensions applied to cell mechanics. *Journal of Computational Physics*, 329:173–209, 2017.
- [12] Chiara Sorgentone, Jeremy I Kach, Aditya S Khair, Lynn M Walker, and Petia M Vlahovska. Numerical and asymptotic analysis of the three-dimensional electrohydrodynamic interactions of drop pairs. *Journal of Fluid Mechanics*, 914, 2021.
- [13] Zydrunas Gimbutas and Leslie Greengard. FMMLIB2D, Fortran libraries for fast multipole methods in two dimensions, 2012–2021. <https://github.com/zgimbutas/fmmlib2d>.
- [14] Dhairya Malhotra and George Biros. PVFMM: A parallel kernel independent FMM for particle and volume potentials. *Commun. Comput. Phys.*, 18(3):808–830, 2015. doi:[10.4208/cicp.020215.150515sw](https://doi.org/10.4208/cicp.020215.150515sw).
- [15] David Shirokoff and J-C Nave. A sharp-interface active penalty method for the incompressible navier–stokes equations. *Journal of Scientific Computing*, 62(1):53–77, 2015.
- [16] David B Stein, Robert D Guy, and Becca Thomases. Immersed boundary smooth extension: a high-order method for solving pde on arbitrary smooth domains using fourier spectral methods. *Journal of Computational Physics*, 304:252–274, 2016.
- [17] David B Stein, Robert D Guy, and Becca Thomases. Immersed boundary smooth extension (ibse): a high-order method for solving incompressible flows in arbitrary smooth domains. *Journal of Computational Physics*, 335:155–178, 2017.
- [18] Fredrik Fryklund, Erik Lehto, and Anna-Karin Tornberg. Partition of unity extension of functions on complex domains. *Journal of Computational Physics*, 375:57–79, 2018.
- [19] Fredrik Fryklund, Mary Catherine A Kropinski, and Anna-Karin Tornberg. An integral equation–based numerical method for the forced heat equation on complex domains. *Advances in Computational Mathematics*, 46(5):1–36, 2020.
- [20] Ludvig af Klinteberg, Travis Askham, and Mary Catherine Kropinski. A fast integral equation method for the two-dimensional navier-stokes equations. *Journal of Computational Physics*, 409:109353, 2020.
- [21] Oscar P Bruno and Mark Lyon. High-order unconditionally stable fc-ad solvers for general smooth domains i. basic elements. *Journal of Computational Physics*, 229(6):2009–2033, 2010.
- [22] Mark Lyon and Oscar P Bruno. High-order unconditionally stable fc-ad solvers for general smooth domains ii. elliptic, parabolic and hyperbolic pdes; theoretical considerations. *Journal of Computational Physics*, 229(9):3358–3381, 2010.
- [23] Oscar P Bruno and Jagabandhu Paul. Two-dimensional fourier continuation and applications. *arXiv preprint arXiv:2010.03901*, 2020.
- [24] Mauro Fontana, Oscar P Bruno, Pablo D Mininni, and Pablo Dmitruk. Fourier continuation method for incompressible fluids with boundaries. *Computer Physics Communications*, 256:107482, 2020.
- [25] Travis Askham and Antoine J Cerfon. An adaptive fast multipole accelerated poisson solver for complex geometries. *Journal of Computational Physics*, 344:1–22, 2017.
- [26] David B Stein, Robert D Guy, and Becca Thomases. Convergent solutions of stokes oldroyd-b boundary value problems using the immersed boundary smooth extension (ibse) method. *Journal of Non-Newtonian Fluid Mechanics*, 268:56–65, 2019.
- [27] Jinzi Mac Huang, Michael J Shelley, and David B Stein. A stable and accurate scheme for solving the stefan problem coupled with natural convection using the immersed boundary smooth extension method. *Journal of Computational Physics*, 432:110162, 2021.
- [28] Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Soc., 2010.
- [29] Alexander H Barnett, Jeremy Magland, and Ludvig af Klinteberg. A parallel nonuniform fast fourier transform library based on an “exponential of semicircle” kernel. *SIAM Journal on Scientific Computing*, 41(5):C479–C504, 2019.
- [30] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, António H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi:[10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- [31] Lineu C Barbosa. A maximum-energy-concentration spectral window. *IBM journal of research and development*, 30(3):321–325, 1986.
- [32] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [33] Kai Hormann and Alexander Agathos. The point in polygon problem for arbitrary polygons. *Computational geometry*, 20(3):131–144, 2001.
- [34] Sean Gillies et al. Shapely: manipulation and analysis of geometric objects, 2007–. URL: <https://github.com/Toblerity/Shapely>.
- [35] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

- [36] Alex Barnett, Bowei Wu, and Shravan Veerapaneni. Spectrally accurate quadratures for evaluation of layer potentials close to the boundary for the 2d stokes and laplace equations. *SIAM Journal on Scientific Computing*, 37(4):B519–B542, 2015.
- [37] Johan Helsing and Rikard Ojala. On the evaluation of layer potentials close to their sources. *Journal of Computational Physics*, 227(5):2899–2921, 2008.
- [38] David B Stein and Alex H Barnett. Quadrature by fundamental solutions: kernel-independent layer potential evaluation for large collections of simple objects. *arXiv preprint arXiv:2109.08802*, 2021.
- [39] Tobin A Driscoll and Nicholas Hale. Rectangular spectral collocation. *IMA Journal of Numerical Analysis*, 36(1):108–132, 2016.
- [40] Sijia Hao, Alex H Barnett, Per-Gunnar Martinsson, and P Young. High-order accurate methods for nyström discretization of integral equations on smooth curves in the plane. *Advances in Computational Mathematics*, 40(1):245–272, 2014.
- [41] Per-Gunnar Martinsson and Vladimir Rokhlin. A fast direct solver for boundary integral equations in two dimensions. *Journal of Computational Physics*, 205(1):1–23, 2005.
- [42] Ankit Rohatgi. Webplotdigitizer: Version 4.5, 2021. URL: <https://automeris.io/WebPlotDigitizer>.
- [43] Isao Hirota and Katsuji Chiyoda. Vector laplacian in general curvilinear coordinates. *The transactions of the Institute of Electrical Engineers of Japan. A*, 102(3):119–126, 1982.
- [44] Y-N Young, Michael J Shelley, and David B Stein. The many behaviors of deformable active droplets. *Mathematical Biosciences and Engineering*, 18(3):2849–2881, 2021.