

A High-order 3D Boundary Integral Equation Solver for Elliptic PDEs in Smooth Domains

Lexing Ying[†], George Biros[‡] and Denis Zorin[‡]

[†] Department of Applied and Computational Mathematics
California Institute of Technology, California 91125

`lexing@acm.caltech.edu`

[‡] Departments of Mechanical Engineering and
Computer and Information Science

University of Pennsylvania, Pennsylvania 19104

`biros@seas.upenn.edu`

[‡] Courant Institute of Mathematical Sciences

New York University, New York 10012

`dzorin@cs.nyu.edu`

October 3, 2005

Abstract

We present a high-order boundary integral equation solver for 3D elliptic boundary value problems on domains with smooth boundaries. We use Nyström’s method for discretization and we combine it with special quadrature rules for the singular kernels that appear in the boundary integrals. The overall asymptotic complexity of our method is $O(N^{3/2})$, where N is the number of discretization points on the boundary of the domain, and corresponds to linear complexity in the number of uniformly sampled evaluation points. A kernel-independent fast summation algorithm is used to accelerate the evaluation of the discretized integral operators. We describe a high-order accurate method for evaluating the solution at arbitrary points inside the domain, including points close to the domain boundary. We demonstrate how our solver, combined with a regular-grid spectral solver, can be applied to problems with distributed sources. We present numerical results for the Stokes, Navier, and Poisson problems.

Keywords. Boundary integral equations, Nyström discretization, singular integrals, nearly singular integrals, fast solvers, Laplace equation, Stokes equation, Navier equation, fast multipole method, fast Fourier transform.

1 Introduction

Potential theory has played a paramount role in both analysis and computation for boundary value problems for elliptic partial differential equations. Numerous applications can be found in fracture mechanics, fluid mechanics, elastodynamics, electromagnetics, and acoustics. Potential theory formulates exterior or interior boundary value problems as integral equations. For problems with known Green’s functions, an integral equation formulation leads to powerful numerical approximation schemes. The advantages of such schemes

are well known: (1) there is no need for volume mesh generation; (2) in many cases they result in operators with bounded condition number; (3) for exterior problems they satisfy far-field boundary conditions exactly; and (4) they typically exhibit high convergence rates when the domain boundary and boundary condition are sufficiently smooth.

Despite their advantages, numerical approximations to integral equations are plagued by several mathematical and implementational challenges, especially if the goal is an algorithm that is asymptotically optimal, accurate, and fast enough to be useful in the practical setting. Indeed, in order to get such an algorithm, for domains with smooth boundaries¹, one has to address five main issues:

- *Fast summation.* The discretized operators are dense and the corresponding linear systems are prohibitively expensive to solve. Direct solvers are not applicable; iterative methods like GMRES can help but still result in suboptimal complexity.
- *Fast and accurate quadratures.* One needs to use suitable quadrature rules to discretize the integral operators; the kernels are often singular or hypersingular, and the choice of the quadrature rule is important to obtain high-order convergence. This is a difficult problem which is made worst by the need to guarantee optimal complexity.
- *Domain boundary representation.* High-accuracy rules often require smooth approximations to the domain boundary. In two dimensions this is a relatively easy problem, but this is not the case in 3D.
- *Solution evaluation.* The solution is typically evaluated on a dense grid of points inside the domain. Such grids can include points arbitrarily close to the boundary, in which case nearly-singular integrals need to be evaluated. Again the goal is to guarantee high accuracy at optimal complexity.
- *Volume potentials.* For problems with possibly highly nonuniform distributed forces one has to devise efficient schemes for the computation of volume integrals, especially in the case that the support of the function coincides with a volumed domain that has a complex boundary.

In this paper we present a method that addresses all issues but the last.

For two-dimensional boundary value problems in smooth domains a number of highly efficient boundary integral solvers have been developed, [5, 21, 23, 32]. Most implementations are based on indirect formulations which result in integral equations with double layer potentials. In 2D these kernels are often non-singular and the domain boundary can be easily parameterized; the boundary integrals can then be evaluated using standard quadrature rules, and superalgebraic convergence rates can be obtained. Such discretization combined with fast summation methods result in optimal algorithms. In three dimensions, however, the situation is radically different. We review the related work in the following section.

We present a 3D boundary integral solver for elliptic PDEs for domains with smooth (C^∞ or C^k -continuous for sufficiently large k , but not necessarily analytic) boundaries, which achieves high-accuracy in linear time in the number of evaluation points. The distinctive features of our solver are: (1) fast kernel-independent summation; (2) arbitrary smooth boundaries and high-order convergence, (3) distributed forces that are uniformly

¹Domains with edges and corners present additional challenges which we do not discuss in this article.

defined in a box that encloses the target domain; and (4) high-accurate direct evaluation of the solution in a non-uniform distribution of points.

The operators are sparsified by our kernel-independent fast multipole method (FMM) [46], which makes it possible to accelerate the solution of the dense linear system for many elliptic PDEs of which the kernels have explicit expressions. We use Nyström’s method to discretize the boundary integral equations. There are two reasons to prefer Nyström’s method to Galerkin or collocation approaches: simpler implementation for superalgebraic convergence and, based in existing literature, lower constants [12].

Although the kernels of various PDEs are different, the behavior of their singularities are similar. We address the second problem (quadrature construction) by extending the local quadrature methods of [11] to integrate the singularities of various types. A key component of the solver is the ability to have high-order representations for arbitrary geometries with (relatively) minimal algorithmic and implementational complexity. Such a representation is described in detail in [48]. To compute the near-singular integrals for points close to the boundary, we adopt a high-order scheme to interpolate the solution from the values at points sufficiently separated from the boundary. Finally, we show how our boundary equation solver can be combined with an FFT-based fast solver on regular grids to obtain solutions of inhomogeneous boundary value problems on domains with arbitrary smooth boundaries.

If N quadrature points are used to discretize the boundary, our solver requires $O(N^{1+\delta})$ operations to solve the boundary integral equation, where $\delta > 0$ is a constant which can be chosen to control the complexity and accuracy of the algorithm. For a Dirichlet problem in which the boundary data is in C^M and the boundary is in C^∞ , the error of the solution is $O(h^{M\delta-(1-\delta)})$, where $h = O(\frac{1}{\sqrt{N}})$ is the spacing of the Nyström discretization. Notice that if both Dirichlet data and the boundary are in C^∞ then δ can be chosen to be arbitrarily close to zero. To simplify the presentation, we describe our solver for $\delta = \frac{1}{2}$ in Section 3.

The complexity of solving the boundary integral equation with $\delta = 1/2$ matches the complexity of evaluating the solution on uniformly sampled volume. Suppose the solution in the interior of the domain is sampled with the same density used for the boundary, i.e. with $O(N^{3/2})$ points, the evaluation of the solution on these domain samples takes $O(N^{3/2})$ operations: in other words, we spend on average a constant number of operations for the evaluation of the solution on each interior point.

1.1 Related work

Much work has been done on using the boundary integral formulation for elliptic problems in two dimensions. We refer the reader to [5, 32, 27] for excellent reviews. Here we primarily focus on work on three-dimensional problems for integral equations of the second kind.

In contrast to the two-dimensional case, almost all boundary integral methods for three-dimensional problems are based on Galerkin or collocation discretizations, with few exceptions, among which [11] is closest to our work. This paper describes an FFT-based method to compute the smooth part of the boundary integral efficiently; a local quadrature rule based on FFT interpolation evaluates the singular part accurately. The only disadvantage of the method (not particularly important for the scattering applications for which it was developed), is that it is not efficient for highly non-uniform domains (for example multiply-connected domains) which make the use of a uniform FFT rather inefficient.

Several types of Galerkin and collocation approaches were explored. Conventional piecewise constant and linear finite element basis functions are often used (e.g. in [5, 10, 13]),

but result in low order convergence.

Higher-order convergence can be obtained with more complex basis functions. Such basis functions, however, are difficult to construct, especially for arbitrary surfaces, and the expense of evaluation of the double integrals for matrix elements is high. The *hybrid* Galerkin method [19] aims to reduce the cost of evaluation of the stiffness matrix. Another common approach is to use a *spectral* Galerkin discretization, e.g. [4, 18, 20]. While excellent convergence rates can be obtained using this method, constructions of spectral basis functions are limited so far to sphere and torus topologies.

Wavelet-based approaches for solving integral equations start with [8]. These approaches have many advantages, such as efficient dense matrix-vector multiplication and preconditioning. An approach using high-order multiwavelets (a wavelet-type basis which is discontinuous but preserves the vanishing moments property) is explored in [1]. In [15, 14, 33], wavelet bases are constructed on the boundary surfaces directly. As it is the case for most Galerkin methods, constructions of high-order basis functions are complex. Consequently, high-order convergence rates for smooth solutions are difficult to obtain. Another wavelet-based approach, based on low-order wavelet bases constructed on a three-dimensional domain containing the boundary was explored in [41, 42, 43]. The important feature of this approach is its ability to handle highly complex and irregular piecewise-smooth geometry.

Finally, p and hp versions of the Galerkin methods [25, 26] were used to obtain high convergence rates for simple open surfaces with corners.

An essential ingredient of Galerkin techniques is an integration method for computing matrix coefficients. A variety of semi-analytical and numerical quadratures for evaluating the integrals for singular and hypersingular kernels were developed e.g. [2, 40, 16, 3]; [28] surveys many of the early algorithms. These algorithms focus on the scenario typical for Galerkin methods when the function to be integrated is known, and are often tailored for specific basis functions. Therefore, most of the quadrature algorithms are not suitable for Nyström formulation since here the basis function is not defined.

A number of techniques were used to accelerate inversion of the dense linear system resulting from discretization. These techniques include fast multipole-based methods [22], panel clustering [24], FFT-based approaches [11, 36] and already mentioned wavelet-based techniques. A comprehensive survey of algorithms using FMM can be found in [37]. For relatively uniform distribution of geometry in space, FFT-based methods are often more efficient.

Nearly singular integration has received relatively little attention; relevant work includes [44, 29, 7]. Vijayakumar *et al.* [44] uses homogeneity of the kernel to convert the problem to an ODE. Johnson [29] uses a change of coordinates to reduce or move the singularity in the parametric domain. Beale and Lai [7] considers the problem in two-dimensions, replacing the kernel with a regularized version, and using correction terms based on asymptotic analysis to reduce errors. A different technique in which the nearly singular evaluation is avoided is presented in [9, 34], in which a jump from the boundary integral equation are combined used to derive discretized monopole and dipole distributions for a regular grid PDE solver. The disadvantages of those methods are that they require regular sampling grids, and that their accuracy is limited by the accuracy of the regular grid solver.

1.2 Boundary value problems

We consider three elliptic equations: the Laplace equation, the Stokes equation, and the Navier equation. The Laplace equation is:

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega \\ u = b & \text{on } \Gamma \end{cases}$$

where u is the potential field. The Stokes equations are

$$\begin{cases} -\mu\Delta \mathbf{u} + \nabla p = 0 & \text{in } \Omega \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega \\ \mathbf{u} = \mathbf{b} & \text{on } \Gamma \end{cases}$$

where \mathbf{u} is the velocity, p is the pressure and μ is a positive constant. Finally, the Navier equation is

$$\begin{cases} -\mu\Delta \mathbf{u} - \frac{\mu}{1-2\nu}\nabla \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega \\ \mathbf{u} = \mathbf{b} & \text{on } \Gamma \end{cases}$$

where \mathbf{u} is the displacement field, μ is a positive constant and $\nu \in (0, \frac{1}{2})$.

We also consider the inhomogeneous form of the above equations, for which we have a distributed force term on the right-hand side.

1.3 Geometric representation of the boundary

To achieve high-order convergence, we need a high-order geometric representation for the boundary. We assume an explicit manifold structure of the boundary, i.e. that the boundary Γ is the union of *overlapping* patches P_k where $k \in K$. Every P_k is parameterized over an open set $U_k \subset \mathbb{R}^2$ by a function $g_k : U_k \rightarrow \mathbb{R}^3$ and $g_k(U_k) = P_k$ (see Figure 1). We assume g_k and the manifold structure to be C^∞ ; this assumption can be relaxed to C^k for a sufficiently large k (how large k should be depends the desired convergence rate). Most of the analytic surfaces, such as spheres, torus, and ellipsoids, have simple manifold-based representations. In [48] we describe a framework to construct general smooth surfaces with explicit manifold representations from arbitrary coarse meshes and we give implementation details.

It is important to note that the surface in this case is *not* defined as a collection of separate and disjoint parameterizations. The points on the surface, quadrature points in particular, are defined as points in 3D, and each point has several images in charts U_k associated with it. Reparametrization techniques (e.g. [30]) can be used to construct a manifold-type representation from an arbitrary fine triangle mesh.

1.4 Contents

In Section 2, we start with a review of the boundary integral formulation of the related elliptic PDE problems, and describe the Nyström's method used for numerical solution. We focus on the Stokes operator. Although it is more complicated than the scalar Laplace operator, it is more general and the derivation of the integral formulations is pertinent to the other kernels. Section 3 describes the quadrature rule for the singular integrals in the boundary integral formulation. Section 4 presents a new algorithm for integrating nearly singular integrals which appear in the evaluation of the solution at points which are close to the boundary. Section 5 shows how to extend our solver to handle inhomogeneous problems with non-zero body force. Finally, numerical results are presented in Section 6.

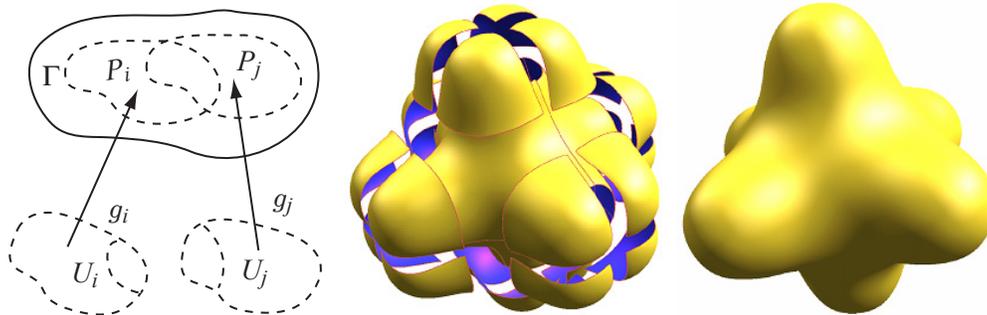


Figure 1: Left: Parameterization of the boundary. The middle and right frames illustrate an example surface. Middle: multiple patches shown separately. Right: the surface is the union of all the patches.

In the rest of the paper, we describe our method in the case of the Stokes equations. For the remaining two equations we discuss the necessary modifications of the method. We limit our discussion to the problems with the Dirichlet boundary conditions.

2 Boundary integral formulation

We use a standard boundary integral formulation of the Stokes equations [35, 31, 38]. The Stokes equations

$$\begin{cases} -\mu\Delta\mathbf{u} + \nabla p = 0 & \text{in } \Omega, \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = \mathbf{b} & \text{on } \Gamma, \end{cases} \quad (1)$$

are converted to a boundary integral equation using the double layer potential for the velocity field:

$$\int_{\Gamma} D(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) \, ds(\mathbf{y}) = \int_{\Gamma} \left(-\frac{3}{4\pi} \frac{(\mathbf{r} \otimes \mathbf{r})(\mathbf{r} \cdot \mathbf{n}(\mathbf{y}))}{|\mathbf{r}|^5} \right) \phi(\mathbf{y}) \, ds(\mathbf{y}) \quad (\mathbf{x} \in \Omega), \quad (2)$$

where $\mathbf{r} = \mathbf{x} - \mathbf{y}$, $\mathbf{n}(\mathbf{y})$ is the exterior normal direction at \mathbf{y} , $|\mathbf{r}|$ is the Euclidean norm of \mathbf{r} ; D is weakly-singular and called the double layer kernel for the velocity field, and the function ϕ , defined on Γ , is called the double layer density. We often write (2) as $(D\phi)(\mathbf{x})$. If we ignore the incompressibility condition for the time being and assume the domain Ω to be simply-connected and bounded, we can obtain the solution of (1) by solving for the density ϕ from

$$\frac{1}{2}\phi(\mathbf{x}) + (D\phi)(\mathbf{x}) = \mathbf{b}(\mathbf{x}) \quad (\mathbf{x} \in \Gamma), \quad (3)$$

and evaluating the velocity \mathbf{u} and pressure p at arbitrary points $\mathbf{x} \in \Omega$ using

$$\mathbf{u}(\mathbf{x}) = \int_{\Gamma} D(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) \, ds(\mathbf{y}) \quad \text{and} \quad p(\mathbf{x}) = \int_{\Gamma} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) \, ds(\mathbf{y}), \quad (4)$$

where $K(\mathbf{x}, \mathbf{y}) = \frac{\mu}{2\pi} \left(\frac{\mathbf{n}(\mathbf{y})}{|\mathbf{r}|^3} - 3\frac{(\mathbf{r} \otimes \mathbf{r})\mathbf{n}(\mathbf{y})}{|\mathbf{r}|^5} \right)$ is the double layer kernel for the pressure field. Then, \mathbf{u} and p for \mathbf{x} on the boundary Γ are given by

$$\mathbf{u}(\mathbf{x}) = \frac{1}{2}[[\mathbf{u}]](\mathbf{x}) + \int_{\Gamma} D(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) \, ds(\mathbf{y}) \quad \text{and} \quad p(\mathbf{x}) = \frac{1}{2}[[p]](\mathbf{x}) + \int_{\Gamma} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) \, ds(\mathbf{y}), \quad (5)$$

where $[[\mathbf{u}]](\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})$ is the difference between the limit values of the velocity fields inside and outside the domain at $\mathbf{x} \in \Gamma$, and $[[p]](\mathbf{x})$ is the difference of the pressure field (see Appendix B for the exact formulae). The integral for pressure in (5) is understood in the *Hadamard* sense.

To account for incompressibility of the fluid and domain geometry we need to modify these basic steps; we consider three domain types shown in Figure 2.

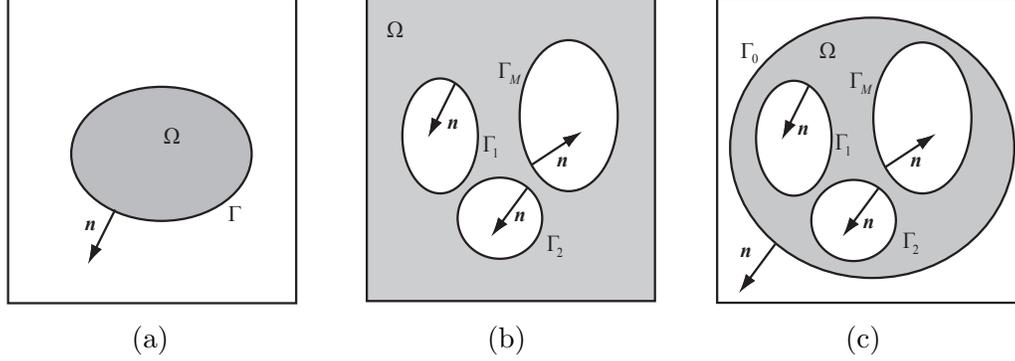


Figure 2: Domain types. (a) single-surface bounded domain, (b) unbounded domain, (c) multiple-surface bounded domain.

Single-surface bounded domains. In this case we need to include the incompressibility condition. We obtain $\boldsymbol{\phi}$ by solving a modified version of (3)

$$\frac{1}{2}\boldsymbol{\phi}(\mathbf{x}) + (D\boldsymbol{\phi})(\mathbf{x}) + (N\boldsymbol{\phi})(\mathbf{x}) = \mathbf{b}(\mathbf{x}) \quad (\mathbf{x} \in \Gamma), \quad (6)$$

where $(N\boldsymbol{\phi})(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \int_{\Gamma} \mathbf{n}(\mathbf{y})\boldsymbol{\phi}(\mathbf{y}) ds(\mathbf{y})$, and then evaluate \mathbf{u} and p by (4).

Unbounded domains. Suppose $\Gamma = \bigcup_{m=1}^M \Gamma_m$ where Γ_m are different connected components of the boundary. In this case, the operator $\frac{1}{2} + D$ has a null space of size $6M$, corresponding to rigid-body transformations. As a result the double layer density $\boldsymbol{\phi}$ is not sufficient to represent arbitrary velocity field in Ω . The solution is to introduce additional *Stokeslet* and *Rotlet* terms for each Γ_m for $1 \leq m \leq M$. The Stokeslet is the Green's function of the Stokes equations:

$$S(\mathbf{r}) = \frac{1}{8\pi\mu} \left(\frac{1}{|\mathbf{r}|} I + \frac{\mathbf{r} \otimes \mathbf{r}}{|\mathbf{r}|^3} \right).$$

The rotlet is defined by

$$R(\mathbf{r})\mathbf{g} = \frac{1}{8\pi\mu} \frac{\mathbf{g} \times \mathbf{r}}{|\mathbf{r}|^3}$$

for any \mathbf{g} . Both are centered at the interior points \mathbf{z}_m of the volumes enclosed by the boundary components. We first solve for $\boldsymbol{\phi}$, α_m and β_m using

$$\left\{ \begin{array}{l} \frac{1}{2}\boldsymbol{\phi}(\mathbf{x}) + (D\boldsymbol{\phi})(\mathbf{x}) + \sum_{m=1}^M (S(\mathbf{x} - \mathbf{z}_m)\alpha_m + R(\mathbf{x} - \mathbf{z}_m)\beta_m) = \mathbf{b}(\mathbf{x}) \quad (\mathbf{x} \in \Gamma), \\ \int_{\Gamma_m} \boldsymbol{\phi}(\mathbf{y}) ds(\mathbf{y}) = 0 \quad (1 \leq m \leq M), \\ \int_{\Gamma_m} (\mathbf{y} - \mathbf{z}_m) \times \boldsymbol{\phi}(\mathbf{y}) ds(\mathbf{y}) = 0 \quad (1 \leq m \leq M). \end{array} \right. \quad (7)$$

Then, the velocity and pressure are computed by

$$\begin{cases} \mathbf{u}(\mathbf{x}) = (D\phi)(\mathbf{x}) + \sum_{m=1}^M (S(\mathbf{x} - \mathbf{z}_m)\alpha_m + R(\mathbf{x} - \mathbf{z}_m)\beta_m), \\ p(\mathbf{x}) = (K\phi)(\mathbf{x}) + \sum_{m=1}^M H(\mathbf{x} - \mathbf{z}_m)\alpha_m, \end{cases} \quad (8)$$

where H is the pressure field associated with the Stokeslet and defined by

$$H(\mathbf{r})\mathbf{g} = \frac{1}{4\pi} \frac{\mathbf{r} \cdot \mathbf{g}}{|\mathbf{r}|^3}.$$

The pressure does not depend on β_m since any rotlet generates zero pressure.

Multiple-surface bounded domains. Suppose the boundary Γ consists of $M + 1$ connected components Γ_m , $m = 0, \dots, M$ and Γ_0 encloses all other components Γ_m for $m = 1, \dots, M$. We first solve for ϕ , α_m and β_m using

$$\begin{cases} \frac{1}{2}\phi(\mathbf{x}) + (D\phi)(\mathbf{x}) + (N\phi)(\mathbf{x}) + \sum_{m=1}^M (S(\mathbf{x} - \mathbf{z}_m)\alpha_m + R(\mathbf{x} - \mathbf{z}_m)\beta_m) = \mathbf{b}(\mathbf{x}) & (\mathbf{x} \in \Gamma), \\ \int_{\Gamma_m} \phi(\mathbf{y}) \, ds(\mathbf{y}) = 0 & (1 \leq m \leq M), \\ \int_{\Gamma_m} (\mathbf{y} - \mathbf{z}_m) \times \phi(\mathbf{y}) \, ds(\mathbf{y}) = 0 & (1 \leq m \leq M). \end{cases} \quad (9)$$

Then, we evaluate the velocity and pressure using (8).

Remark 2.1. The integral equations for the Navier equations are very similar for the Stokes equations. The only difference is that $N\phi$ is removed from the equations of the single-surface and multiple-surface bounded domains, since there is no incompressibility condition.

The integral equations of the Laplace equation are simpler. For single-surface bounded domains (3) works without modification. For both the unbounded and multiple-surface bounded cases, the equations are

$$\begin{cases} \frac{1}{2}\phi(\mathbf{x}) + (D\phi)(\mathbf{x}) + \sum_{m=1}^M S(\mathbf{x} - \mathbf{z}_m)\alpha_m = \mathbf{b}(\mathbf{x}) & (\mathbf{x} \in \Gamma), \\ \int_{\Gamma_m} \phi(\mathbf{y}) \, ds(\mathbf{y}) = 0 & (1 \leq m \leq M), \end{cases}$$

where S is the Green's function of the Laplace equation and ϕ and α_m are the unknowns.

Returning to the Stokes equations, we discretize (6), (7) or (9) using the Nyström method, and the resulting system is solved by means of a GMRES solver and a preconditioner described in [21]. The essential step in the GMRES solver is the evaluation of

$$(D\phi)(\mathbf{x}) = \int_{\Gamma} D(\mathbf{x}, \mathbf{y})\phi(\mathbf{y}) \, ds(\mathbf{y}) \quad (\mathbf{x} \in \Gamma). \quad (10)$$

This integral needs to be computed efficiently as it is evaluated at each iteration of the GMRES solver, and also accurately as its accuracy determines the overall accuracy of the method.

3 Numerical quadrature

In our discussion of the quadrature rules for integrals (10) the domain boundary Γ is the union of a set of *overlapping* patches P_k , $k = 1, \dots, K$, smoothly parameterized over a chart U_k by smooth functions g_k (from now on, we use the word *smooth* to describe C^∞ or C^k functions for k sufficiently large not to affect the convergence estimates). Additionally, we utilize a partition of unity $w_k : \Gamma \rightarrow \mathbb{R}$ satisfying the following conditions:

- w_k is smooth, non-negative, and supported in the interior of P_k ,
- $\sum_{k=1}^K w_k(\mathbf{x}) = 1$ for all $\mathbf{x} \in \Gamma$.

3.1 Discretization and quadrature rules

As we integrate singular kernels, we need to use high-accuracy quadrature rules for singular functions. Commonly used adaptive methods with variable number of quadrature points combined with product integration rules can be used in the context of the Nyström method, these techniques are difficult to combine with fast summation schemes. A way to do this is to choose regular quadrature point placement, and take advantage of domain transformations to adapt quadrature points to the singularity. Our approach is based on the approach developed in [11] for the Helmholtz equation.

We first restrict the integral $(D\phi)(\mathbf{x})$ to the domain U_k :

$$(D\phi)(\mathbf{x}) = \sum_{k=1}^K \int_{U_k} D(\mathbf{x}, g_k(c_k)) w_k(g_k(c_k)) \phi(g_k(c_k)) J_k(c_k) dc_k,$$

where J_k denotes the determinant of the Jacobian matrix of the parametrization g_k . Define

$$\psi_k(c_k) = w_k(g_k(c_k)) \phi(g_k(c_k)) J_k(c_k),$$

which vanishes at the boundary of U_k , since w_k is vanishing. Using this notation, the integral becomes

$$(D\phi)(\mathbf{x}) = \sum_{k=1}^K \int_{U_k} D(\mathbf{x}, g_k(c_k)) \psi_k(c_k) dc_k. \quad (11)$$

We use a subset of a regularly spaced grid of quadrature points $\{(ah, bh), a, b \in \mathbb{Z}$ for each domain U_k . The set of grid points inside U_k is denoted by $\{c_{k,i}\}$. We use the union of grid points in all charts $\bigcup_{k=1}^K \{c_{k,i}\}$ as the Nyström points in the approximation of $D\phi(\mathbf{x})$.

In subsequent formulas, we use $\phi_{k,i}$ to denote the value of function ϕ at points $c_{k,i}$; similarly, $\psi_{k,i}$ is the value of ψ_k at $c_{k,i}$ and $\mathbf{x}_{k,i}$ is the position $g_k(c_{k,i})$.

We consider one by one the integrals in (11) over a single domain U_k . To simplify the notation, we drop the index k in the rest of this section, i.e. we compute integrals of the form

$$\int_U D(\mathbf{x}, g(c)) \psi(c) dc. \quad (12)$$

Non-singular integration. If $\mathbf{x} \notin g(U)$, then $D(\mathbf{x}, g(c))$ is non-singular for any point $c \in U$, and the integrand $D(\mathbf{x}, g(c))\psi(c)$ and its derivatives vanish at the boundary of U . Such a function can be extended to a smooth periodic function on the plane, by extending it by zero to a rectangular domain containing U . For infinitely differentiable periodic

functions, the trapezoidal rule with weights h^2 at points c_i has super-algebraic rate of convergence for integrals over the periodic domain. For C^M functions, the trapezoidal rule converges as $O(h^M)$.

Singular integration. If $\mathbf{x} = g(c')$ for some $c' \in U$, we introduce a C^∞ function $\eta_{c'}$ defined by

$$\eta_{c'}(c) = \chi\left(\frac{|c - c'|}{\sqrt{h}}\right)$$

where $\chi : [0, \infty) \rightarrow [0, 1]$ is a non-increasing C^∞ function satisfying $\chi(r) = 1$ in the neighborhood of zero and $\chi(r) = 0$ for $r \geq 1$. In practice, we choose χ such that $\chi(r) = 1$ for $r \leq 1/4$. The function $\eta_{c'}$ is radially-symmetric and supported in a disk of radius \sqrt{h} centered at c' . Following [11], we call $\eta_{c'}$ a *floating partition of unity* function since its support depends on the position of \mathbf{x} . Using the function $\eta_{c'}$, we split the integral (12) into two parts:

$$\int_U D(g(c'), g(c)) \psi(c) dc = \int_U D(g(c'), g(c)) (1 - \eta_{c'}(c)) \psi(c) dc \quad (13)$$

$$+ \int_U D(g(c'), g(c)) \eta_{c'}(c) \psi(c) dc. \quad (14)$$

The integrand of (13) is smooth because $(1 - \eta_{c'}(c))$ vanishes in the neighborhood of c' , and can be integrated using the trapezoidal rule. To integrate (14), we use a polar coordinate system centered at c' . Let $q = c - c' = (\rho \cos(\theta), \rho \sin(\theta))$. Changing the variables in the second integral we obtain

$$\int_0^\pi d\theta \int_{-\sqrt{h}}^{\sqrt{h}} \rho D(g(c'), g(c(\rho, \theta))) \chi\left(\frac{|\rho|}{\sqrt{h}}\right) \psi(c(\rho, \theta)) d\rho \quad (15)$$

In this formula, we allow ρ to be negative by restricting the integration domain of θ to be $[0, \pi)$. It can be shown (see Appendix A) that for a fixed θ , $\rho D(g(c'), g(c(\rho, \theta)))$ is a smooth function of ρ . Since $\chi(\cdot)$ vanishes at 1, the integrand of the inner integral can be extended to a 1D smooth periodic function. Moreover, since the inner integral depends smoothly and periodically on θ , the double integral (15) can be integrated using trapezoidal rule again. The coordinates of the quadrature points in the polar coordinate system (ρ, θ) are

$$\left(\left(a - \frac{1}{2}\right)h, 2\pi b / \left\lceil \frac{2\pi}{\sqrt{h}} \right\rceil \right) \in [-\sqrt{h}, \sqrt{h}] \times [0, \pi)$$

where a and b range over all integers for which the points are in the integration domain (Figure 3). The number of quadrature points is of order $O(\frac{1}{h})$.

The proof of the high-order accuracy of this quadrature rule is given in Theorem A.3 of Appendix A. Assuming the boundary surface and the boundary conditions are both infinitely differentiable, the quadrature rule is of order $O(h^M)$ for any integer M .

Density interpolation. The quadrature points for (14) are on a Cartesian grid in polar coordinates which do not coincide with the points at which the unknown function is discretized. An interpolation procedure is necessary to obtain the values of ψ at the polar-coordinate quadrature points from its values at the Cartesian quadrature points. The method we use, closely related to the technique used in [11], has two steps: the preprocessing step and the evaluation step.

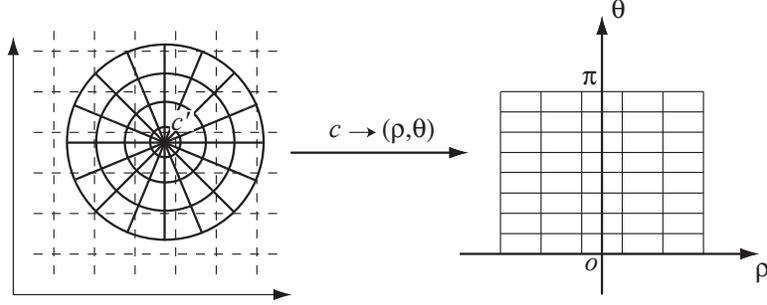


Figure 3: Integration points in Cartesian (left) and polar (right) coordinates.

In the preprocessing step, we compute an additional set of values of ψ on the chart U . Since ψ can be extended to a smooth periodic function on the plane with the periodic domain being a rectangle containing regularly spaced quadrature points c_i , we use 2D Fast Fourier transform (FFT) to calculate the Fourier coefficients of the periodic extension of ψ from the values ψ_i at c_i . We then use these Fourier coefficients to approximate the value of ψ on a new grid, which is m times finer than the original grid by means of an inverse FFT.

Given a point $c \in U$, we need to interpolate the values of ψ at a polar-coordinate grid centered at c . This is done by using the bicubic B-spline interpolation with periodic end conditions on the refined grid.

This interpolation procedure is quite efficient due to local support of ψ which makes it possible to extend it periodically and use FFT. By choosing the fine grid resolution sufficiently high, we can obtain arbitrarily small *relative* error bound ϵ_{interp} for the interpolation process. In practice, for $m = 8$ we obtain $\epsilon_{interp} = 10^{-8}$.

Our quadrature rules have two important novel aspects: (1) the choice of the size of the support of the floating partition of unity function η is of size \sqrt{h} which enables us to prove error bounds and complexity results for the numerical integration; (2) the integration grid for the singularity is fully symmetric in the polar coordinates (see Figure 3), which makes it possible to compute hypersingular integrals such as the one in the evaluation of the pressure of the Stokes equations (see Appendix A).

3.2 Efficient implementation and complexity analysis

Integral evaluation with the quadrature rules of Section 3.1 can be implemented efficiently without compromising the accuracy by using the kernel independent fast multipole method.

For each point \mathbf{x} on the boundary Γ , we need to evaluate the integral

$$(D\phi)(\mathbf{x}) = \sum_{k=1}^K \int_{U_k} D(\mathbf{x}, g_k(c_k)) \psi_k(c_k) dc_k,$$

which is decomposed into the sum of three parts:

$$\sum_{k: \mathbf{x} \notin g_k(U_k)} \int_{U_k} D(\mathbf{x}, g_k(c_k)) \boldsymbol{\psi}_k(c_k) dc_k, \quad (16)$$

$$\sum_{k: \mathbf{x} \in g_k(U_k)} \int_{U_k} D(\mathbf{x}, g_k(c_k)) (1 - \eta_{c'_k}(c_k)) \boldsymbol{\psi}_k(c_k) dc_k, \quad (17)$$

$$\sum_{k: \mathbf{x} \in g_k(U_k)} \int_{U_k} D(\mathbf{x}, g_k(c_k)) \eta_{c'_k}(c_k) \boldsymbol{\psi}_k(c_k) dc_k, \quad (18)$$

where $c'_k = g_k^{-1}(\mathbf{x})$ is a point in the chart U_k . The integral (18) is evaluated using the trapezoidal rule in local polar coordinates centered at c'_k . The integrals (16) and (17) are non-singular and are evaluated by means of the standard trapezoidal rule applied to $c_{k,i}$. The discretization of these two integrals is given by

$$\sum_{\text{all } k} \sum_i D(\mathbf{x}, g_k(c_{k,i})) \boldsymbol{\psi}_{k,i} h^2 - \sum_{k: \mathbf{x} \in g_k(U_k)} \sum_i D(\mathbf{x}, g_k(c_{k,i})) \eta_{c'_k}(c_{k,i}) \boldsymbol{\psi}_{k,i} h^2. \quad (19)$$

where $\boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_k(c_{k,i})$. In the first summation extending over the whole boundary surface, the terms $\boldsymbol{\psi}_{k,i} h^2$ are independent of the evaluation point \mathbf{x} . Therefore, we can use the kernel independent fast multipole method developed in [46] to evaluate the value for all quadrature points \mathbf{x} efficiently without compromising the accuracy. The second summation only involves the points $c_{k,i}$ for which $\eta_{c'_k}$ is positive. As the support of $\eta_{c'_k}$ has radius \sqrt{h} , the absolute number of such points grows as the discretization is refined, but the *fraction* of these points is $O(h)$ and thus approaches zero. Algorithm 1 is the pseudo-code for evaluation of all $\mathbf{d}_{l,j} = (D\phi)(\mathbf{x}_{l,j})$, where indices k and l range over all charts, and indices i and j over all Cartesian grid quadrature points within each chart.

Algorithm 1 Singular integral evaluation for velocity (Stokes equation).

```

for all  $(k, i)$  do
   $\boldsymbol{\psi}_{k,i} \leftarrow w_{k,i} \boldsymbol{\phi}_{k,i} J_k(c_{k,i})$ 
end for
{Step 1: Add terms of (19) with evaluation point independent weights.}
Set all  $\mathbf{d}_{l,j}$  to approximation of  $\sum_{k,i} D(\mathbf{x}_{l,j}, \mathbf{x}_{k,i}) \boldsymbol{\psi}_{k,i} h^2$  using kernel-independent FMM.
{Step 2: Subtract terms of (19) with evaluation point dependent weights.}
for all  $(l, j)$  do
  for each  $U_k$  such that  $\mathbf{x}_{l,j} = g_k(c'_k)$  for some  $c'_k \in U_k$  do
     $\mathbf{d}_{l,j} \leftarrow \mathbf{d}_{l,j} - \sum_{i: \boldsymbol{\psi}_{k,i} > 0} D(\mathbf{x}_{l,j}, g_k(c_{k,i})) \eta_{c'_k}(c_{k,i}) \boldsymbol{\psi}_{k,i} h^2.$ 
  end for
end for
{Step 3: Preprocess the grids  $\boldsymbol{\psi}_{k,i}$  for high-order interpolation.}
{Step 4: Add (18).}
for all  $(l, j)$  do
  for all  $U_k$  such that  $\mathbf{x}_{l,j} = g_k(c'_k)$  for some  $c'_k \in U_k$  do
    Add to  $\mathbf{d}_{l,j}$  the discretization of  $\int_{U_k} D(\mathbf{x}_{l,j}, g_k(c_k)) \eta_{c'_k}(c_k) \boldsymbol{\psi}_k(c_k) dc_k$  using polar coordinates integration.
  end for
end for

```

Complexity analysis. Let N be the total number of quadrature points $\mathbf{x}_{k,i}$. This number can be approximated by $K/h^2 = O(1/h^2)$, where K is the total number of patches covering the boundary surface Γ which does not depend on h . The total computational cost is the sum of the costs of four stages of the algorithm.

1. The kernel-independent FMM algorithm used at Step 1 has complexity $O(N)$ for a prescribed error ϵ_{FMM} .
2. In the double **for** loop of Step 2, since the support of the floating partition of unity is a disk of radius \sqrt{h} , the number of quadrature points $\mathbf{x}_{k,i}$ at which the evaluation is required is $O(\sqrt{N})$. Therefore, the overall complexity $O(N^{3/2})$.
3. The preprocessing of Step 3 has complexity of the fast Fourier transform, $O(N \log N)$
4. The double **for** loop of Step 4, is also $O(N^{3/2})$ since the number of polar-coordinate quadrature points is of order $O(\frac{1}{h}) = O(\sqrt{N})$.

Summing up all stages of the algorithm, we observe that complexity of our quadrature algorithm is $O(N^{3/2})$. This complexity is mostly determined by the radius of the floating partition of unity. The error analysis is carried out in Appendix A and states that the error is $O(h^{\frac{M-1}{2}})$ if the double layer density ϕ is C^M . If instead of using the radius proportional to \sqrt{h} we use the radius $h^{1-\gamma}$ for $0 < \gamma \leq 1$, we obtain an algorithm with complexity $N^{1+\gamma}$. For example, by using a partition of unity which shrinks faster (e.g. as $h^{3/4}$), we can speed up the algorithm at the cost of letting the error decrease more slowly with h , i.e. lowering the approximation order. On the contrary, by using a partition of unity which shrinks slower (e.g. $h^{1/4}$), we can increase the approximation order, but the complexity of each evaluation step increases.

The algorithm described above implements a linear operator mapping the vector of $\phi_{k,i}$ to the vector of $\mathbf{d}_{l,j}$. We will use Y^D to denote this linear operator in the rest of the paper. To summarize,

- Evaluation of Y^D has complexity $O(N^{3/2})$.
- The approximation error $(Y^D \phi - D\phi)(\mathbf{x})$, when ϕ is a C^M function, is bounded by $\max(C_1 h^{\frac{M-1}{2}}, C_2 \epsilon_{FMM}, C_2 \epsilon_{interp})$ for \mathbf{x} on Γ , where the constant C_1 depends on the M th order derivatives of ϕ and C_2 is a bound on the L^∞ norm of $D\phi$. As we mentioned earlier, ϵ_{FMM} is the error bound of the kernel independent FMM and ϵ_{interp} the error bound of the interpolation step.

Remark 3.1. For the Laplace equation, the algorithm described in this section can be used without modifications since the kernel has the same singularity behavior as the one for the Stokes equations. For the Navier equation the double layer kernel has stronger singularity and the whole integral is understood in the Cauchy sense. Nevertheless, based on an argument similar to Theorem A.6 of Appendix A, the trapezoidal rule can still be applied, and no changes in the algorithm are necessary in this case either.

3.3 Hypersingular pressure evaluation

In this section, we describe the algorithm to evaluate the pressure value p on the boundary Γ from the solution ϕ using (5). This is also an essential step in our algorithm for the evaluation of \mathbf{u} and p in the interior of the domain Ω (see Section 4).

Let ϕ be the double layer density on Γ . For $\mathbf{x} \in \Omega$, the double layer representation for pressure $p(\mathbf{x})$ is

$$p(\mathbf{x}) = (K\phi)(\mathbf{x}) = \int_{\Gamma} \frac{\mu}{2\pi} \left(\frac{\mathbf{n}(\mathbf{y})}{|\mathbf{r}|^3} - 3 \frac{(\mathbf{r} \otimes \mathbf{r})\mathbf{n}(\mathbf{y})}{|\mathbf{r}|^5} \right) \cdot \phi(\mathbf{y}) \, ds(\mathbf{y}),$$

where $\mathbf{r} = \mathbf{x} - \mathbf{y}$ and $\mathbf{n}(\mathbf{y})$ is the exterior normal direction. We use K to denote both the kernel and the integral operator. The kernel of K is fundamentally different from other kernels that we consider, as the singularity of the kernel is of order $|\mathbf{r}|^{-3}$, i.e. the integral is hypersingular. Evaluating these integrals requires modification of our algorithm.

In order to derive the formula for $p(\mathbf{x})$ for \mathbf{x} on the boundary Γ , we use the following fact (from potential theory for the Stokes operator): if $\phi \equiv c$ is a constant, then the velocity field \mathbf{u} in Ω generated by ϕ is again a constant, and correspondingly, the pressure field p is zero [32, 38, 39]. Suppose $\mathbf{x}' \in \Omega$ approaches a boundary point $\mathbf{x} \in \Gamma$, then we can write the formula for the pressure in the following form:

$$p(\mathbf{x}') = \int_{\Gamma} K(\mathbf{x}', \mathbf{y}) \cdot (\phi(\mathbf{y}) - \phi(\mathbf{x})) \, ds(\mathbf{y}),$$

If \mathbf{x}' were be on the boundary Γ , these integrals would be interpreted in the Cauchy sense. We know that for this type of integrals, the interior limit $p(\mathbf{x})$ of $p(\mathbf{x}')$ has the following integral form:

$$p(\mathbf{x}) = \frac{1}{2} [[p]](\mathbf{x}) + \int_{\Gamma} K(\mathbf{x}, \mathbf{y}) \cdot (\phi(\mathbf{y}) - \phi(\mathbf{x})) \, ds(\mathbf{y}), \quad (20)$$

where $[[p]]$ is the difference between the interior limit and the exterior limit of p at \mathbf{x} .

The jump $[[p]](\mathbf{x})$ can be expressed in terms of the double layer density ϕ . We choose a local orthonormal frame $\boldsymbol{\alpha}, \boldsymbol{\beta}$ in the tangent plane at \mathbf{x} . In Appendix B, we show that the jump for p is given by

$$[[p]] = -2\mu(\boldsymbol{\alpha}^t \phi_{\boldsymbol{\alpha}} + \boldsymbol{\beta}^t \phi_{\boldsymbol{\beta}})$$

where $\phi_{\boldsymbol{\alpha}}$ and $\phi_{\boldsymbol{\beta}}$ denote the directional derivatives of ϕ in the directions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively.

Jump evaluation. To evaluate the jumps $[[p]](\mathbf{x})$, we need to evaluate $\phi_{\boldsymbol{\alpha}}$ for a direction of $\boldsymbol{\alpha}$ in the tangent plane at \mathbf{x} from the values of ϕ at the quadrature points $\mathbf{x}_{k,i}$. The most straightforward approach would be to use differential of the chart parametrization to map directions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to the parametric domain, evaluate the directional derivatives at quadrature points, and interpolate in the parametric domain.

However, since the density ϕ is not compactly supported on each parametric domain, directly interpolating the directional derivatives of ϕ often results low order accuracy at points close to the boundary of the parametric domain. To achieve high-order approximation to the directional derivatives $\phi_{\boldsymbol{\alpha}}$ and $\phi_{\boldsymbol{\beta}}$, we use the partition of unity again. Using $\phi_{\boldsymbol{\alpha}}$ as an example, we write

$$\phi_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{k:\mathbf{x} \in g_k(U_k)} (w_k \phi)_{\boldsymbol{\alpha}}(\mathbf{x}).$$

Here $w_k \phi$ are compactly supported functions in domains U_k and can be extended periodically. Therefore, on each U_k we can use an interpolation procedure similar to the one we developed for interpolating ψ . Again, an FFT-based preprocessing step is used to build an eight-fold refined grid for interpolation of each directional derivative. At the evaluation stage, we use the B-spline interpolation to approximate the value of $(w_k \phi)_{\boldsymbol{\alpha}}$.

Singular integral evaluation. Using Theorem A.6 we can accurately evaluate the integral in (20) by using the numerical integration operator Y^K (introduced in Section 3.2) on the double layer density $\phi - \phi^{\mathbf{x}}$, where $\phi^{\mathbf{x}}(\mathbf{y})$ is a constant density with value $\phi(\mathbf{x})$ for any $\mathbf{y} \in \Gamma$. However, $\phi - \phi^{\mathbf{x}}$ depends on the target point \mathbf{x} , and the result $Y^K(\phi - \phi^{\mathbf{x}})$ only gives the valid pressure value at the point \mathbf{x} . Clearly applying the operator Y^K to $\phi - \phi^{\mathbf{x}}$ for each \mathbf{x} is prohibitively expensive. The algorithm we propose uses the linearity of Y^K to evaluate $(K\phi)(\mathbf{x})$ at all points \mathbf{x} simultaneously in a much more efficient way.

Observe that

$$\begin{aligned} (Y^K(\phi - \phi^{\mathbf{x}}))(\mathbf{x}) &= (Y^K\phi)(\mathbf{x}) - (Y^K\phi^{\mathbf{x}})(\mathbf{x}) \\ &= (Y^K\phi)(\mathbf{x}) - ((Y^K\mathbf{e}^1)(\mathbf{x}), (Y^K\mathbf{e}^2)(\mathbf{x}), (Y^K\mathbf{e}^3)(\mathbf{x})) \cdot \phi(\mathbf{x}) \end{aligned}$$

Algorithm 2 summarizes numerical integration of $K\phi(\mathbf{x})$ for a set of points \mathbf{x} on Γ , where \mathbf{e}^1 , \mathbf{e}^2 and \mathbf{e}^3 are the constant double layer densities on Γ with values $(1, 0, 0)^t$, $(0, 1, 0)^t$ and $(0, 0, 1)^t$ respectively. Note that although $K\phi$ is not defined for arbitrary smooth ϕ (it is only defined for a smooth function ϕ which vanishes at \mathbf{x}), $Y^K\phi$ is defined for all points \mathbf{x} as a *numerical* integration operator.

Algorithm 2 Singular integration for the pressure.

Evaluate $g^d = Y^K\mathbf{e}^d$ for $d = 1, 2, 3$.

Evaluate $p = Y^K\phi$.

for each evaluation point \mathbf{x} **do**

$p(\mathbf{x}) \leftarrow p(\mathbf{x}) - (g^1(\mathbf{x}), g^2(\mathbf{x}), g^3(\mathbf{x})) \cdot \phi(\mathbf{x})$.

end for

Since the operator Y^K has complexity $O(N^{3/2})$, this algorithm also has complexity $O(N^{3/2})$. For a fixed combination of evaluation points and Nyström discretization points, the first step of the algorithm needs to be done only once, and can be reused for different double layer density function ϕ . As the discretization is refined, convergence of the approximation to the correct values depends on cancellation: both values $Y^K\phi$ and $Y^K\phi^{\mathbf{x}}$ at point \mathbf{x} may increase, however, their difference approaches the correct limit value. This indicates that the algorithm may potentially suffer from floating point errors due to the cancellation of large quantities. In our numerical experiments we have not observed the degradation of the accuracy.

4 Nearly singular integration

In this section, we present the algorithm to evaluate the velocity \mathbf{u} and the pressure p at an arbitrary point \mathbf{x} in the domain. We explain the algorithm for velocity \mathbf{u} , and comment on the algorithm for p at the end of this section.

For $\mathbf{x} \in \Omega$, the expression for \mathbf{u} is

$$\mathbf{u}(\mathbf{x}) = (D\phi)(\mathbf{x}) = \int_{\Gamma} D(\mathbf{x}, \mathbf{y})\phi(\mathbf{y}) \, d\mathbf{y}.$$

The integrand is not singular since $D(\mathbf{x}, \mathbf{y})$ is not singular when $\mathbf{x} \notin \Gamma$. If the distance from \mathbf{x} to Γ is bounded from below by a positive constant, we can bound the derivatives of the integrand. In this case the trapezoidal rule on each chart $U_k, k = 1, \dots, K$ with evenly spaced quadrature points $c_{k,i}$ has optimal accuracy. However, as \mathbf{x} approaches the

boundary Γ , $D(\mathbf{x}, \mathbf{y})$ becomes nearly singular and oscillatory, and the derivatives of $D(\mathbf{x}, \mathbf{y})$ with respect to \mathbf{y} cannot be bounded uniformly.

We propose an algorithm that evaluates $(D\phi)(\mathbf{x})$ at any point $\mathbf{x} \in \Omega$ from the values at quadrature points on the boundary Γ with high-order accuracy, no matter how close \mathbf{x} is to the boundary.

The idea of our algorithm is to partition Ω into three regions and use different schemes for integral evaluation within each region. Given the discretization spacing h , we partition the domain Ω into the following regions (see Figure 4(a)):

- Well-separated: $\Omega_0 = \{\mathbf{x} \in \Omega \mid \text{dist}(\mathbf{x}, \Gamma) \in (\sqrt{h}, \infty)\}$,
- Intermediate: $\Omega_1 = \{\mathbf{x} \in \Omega \mid \text{dist}(\mathbf{x}, \Gamma) \in (h, \sqrt{h}]\}$,
- Nearest: $\Omega_2 = \{\mathbf{x} \in \Omega \mid \text{dist}(\mathbf{x}, \Gamma) \in (0, h]\}$.

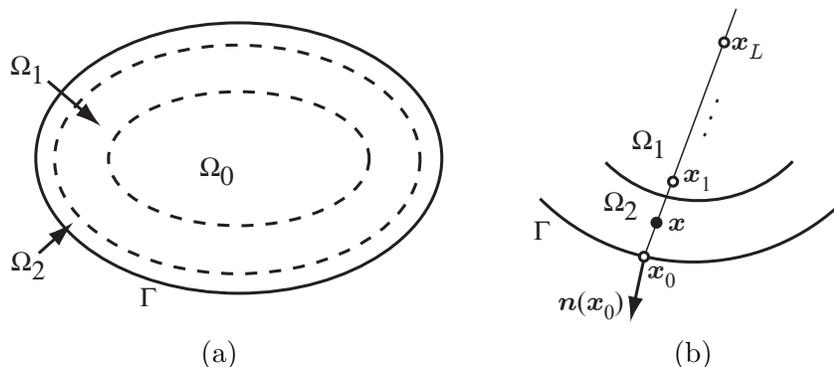


Figure 4: Evaluation of nearly singular integrals. (a): different regions based on their distance to the boundary. (b): evaluation procedure for $\mathbf{x} \in \Omega_2$.

If \mathbf{x} is in the *well-separated* region Ω_0 , we use the trapezoidal rule on the Nyström points $\mathbf{x}_{k,i}$ with weights $\psi_{k,i}$ to evaluate $(D\phi)(\mathbf{x})$, where $\psi_{k,i}$ are defined as in Section 3.2.

For the points \mathbf{x} in the *intermediate* region Ω_1 , for each chart U_k we resample the function ψ_k on a refined Cartesian grid with spacing $h^{3/2}$ using the values $\psi_{k,i}$ at the original grid points $\mathbf{x}_{k,i}$ with grid step h . Since both grids are Cartesian and ψ_k is periodic function on U_k , the resampling can be done by first computing FFT of ψ_k , then padding the higher frequency with zeros, and finally using inverse FFT to compute the values on the refined grid. With ψ_k 's values on the refined grid available, we use the trapezoidal rule on the new grid to evaluate $D\phi(\mathbf{x})$. The denser sampling with spacing $h^{3/2}$ ensures that the approximation order of the quadrature rule is maintained (Appendix C).

Finally, for points in the *nearest* region Ω_2 , we interpolate between values at points on the surface and points in Ω_1 or Ω_0 . For each point \mathbf{x} in Ω_2 , we first find a point $\mathbf{x}_0 \in \Gamma$ such that

$$\frac{\mathbf{x}_0 - \mathbf{x}}{|\mathbf{x}_0 - \mathbf{x}|} \cdot \mathbf{n}(\mathbf{x}_0) \geq \alpha, \quad (21)$$

where the value of α is less than but close to 1, which means that $\mathbf{x} - \mathbf{x}_0$ is almost orthogonal to the tangent plane at \mathbf{x}_0 . The algorithm is not very sensitive to the value of α ; we use $\alpha = 0.95$ in our implementation. We find \mathbf{x}_0 using a Newton-type nonlinear iteration to maximize the dot product left-hand side of (21). In most cases, the nonlinear solver finds such a point \mathbf{x}_0 in 3 or 4 iterations.

We then define points $\{\mathbf{x}_l, l = 1, \dots, L\}$ by

$$\mathbf{x}_l = \mathbf{x}_0 + l \cdot \frac{\mathbf{x} - \mathbf{x}_0}{|\mathbf{x} - \mathbf{x}_0|} \beta h,$$

where β is a constant such that $\alpha \beta$ is greater than but close to 1. Since α is close to 1, we can choose β to be close to 1 as well. We then use the singular quadrature algorithm described in Section 3 to evaluate $\frac{1}{2}\phi(\mathbf{x}_0) + D\phi(\mathbf{x}_0)$ which is the limit of $D\phi$ at \mathbf{x}_0 . The points $\{\mathbf{x}_l, l = 1, \dots, L\}$ are now in Ω_1 or Ω_0 since $\alpha\beta \geq 1$, and we evaluate $\{(D\phi)(\mathbf{x}_l), l = 1, \dots, L\}$ using the procedure described above for regions Ω_1 and Ω_0 . Finally, we use these values at $\{\mathbf{x}_l, l = 0, \dots, L\}$ to perform a 1D Lagrange interpolation of order L to obtain the value of $D\phi$ at \mathbf{x} (see Figure 4(b)). The order of interpolation is chosen to achieve the desired convergence rate, as discussed in Appendix C.

Summary and complexity analysis. To evaluate the velocity on $O(N^{3/2})$ points that uniformly sample Ω with approximately the same discretization used for the boundary we first identify for each point which region (Ω_0, Ω_1 or Ω_2) it belongs to and then we use the corresponding algorithm described in the previous section.

- There are $O(N^{3/2})$ points in the well-separated region Ω_0 . We evaluate the velocity using trapezoidal rule on the quadrature points $\mathbf{x}_{k,i}$ with FMM acceleration. This step has complexity $O(N^{3/2})$.
- There are $O(N^{5/4})$ points in intermediate region Ω_1 . We first interpolate ϕ onto a refined grid with spacing $h^{3/2}$ and evaluate the potential using the trapezoidal rule on the refined quadrature points with again FMM acceleration. The complexity of this step is also $O(N^{3/2})$.
- There are $O(N)$ points in nearest region Ω_2 . For each point \mathbf{x} in Ω_2 , we first find the correspondent \mathbf{x}_0 and \mathbf{x}_l for $l = 1, \dots, L$. The evaluation of $\frac{1}{2}\phi(\mathbf{x}_0) + (D\phi)(\mathbf{x}_0)$ for all points $\{\mathbf{x}_0 : \mathbf{x} \in \Omega_2\}$ can be done using the algorithm in Section 3. We then evaluate $\{(D\phi)(\mathbf{x}_l) : \mathbf{x} \in \Omega_2, l = 1, \dots, L\}$ using the refined grid with FMM acceleration. The complexity of the evaluation for $\{\mathbf{x} \in \Omega_2\}$ is $O(N^{3/2})$ again.

We conclude that the complexity of the complete algorithm is $O(N^{3/2})$.

In Appendix C we prove that the high-order accuracy of the algorithm. Assuming the boundary surface and the Dirichlet data are infinitely smooth, the error of the evaluation is $O(h^L)$, which is governed by the order L of the 1D Lagrange interpolation.

The algorithm to evaluate the pressure p at arbitrary point \mathbf{x} in Ω is similar to the presented algorithm for \mathbf{u} . The only difference is that since p has a high-order singularity, the error bound for the velocity of points in regions Ω_1 and Ω_2 decreases slower with h , and it is observed in practice. To improve the accuracy, we use singularity subtraction: for a point \mathbf{x} in Ω_1 or Ω_0 which is close to Γ , we find its nearest point \mathbf{x}_0 on the boundary. Then instead of evaluating the pressure using $\int_{\Gamma} K(\mathbf{x}, \mathbf{y})\phi(\mathbf{y}) ds(\mathbf{y})$, we use

$$\int_{\Gamma} K(\mathbf{x}, \mathbf{y})(\phi(\mathbf{y}) - \phi^{\mathbf{x}_0}(\mathbf{y})) ds(\mathbf{y}),$$

where $\phi^{\mathbf{x}_0}(\mathbf{y}) = \phi(\mathbf{x}_0)$ for any \mathbf{y} . Similarly to the case of singular quadrature the order of singularity is reduced. Efficient implementation for the second integral follows the ideas in Section 3.3.

Remark 4.1. The algorithms for nearly singular integration for the Laplace equation and the Navier equation are exactly the same.

5 Inhomogeneous Stokes equations

In the preceding sections we have focused on the *homogeneous* Dirichlet problem for the Stokes equations:

$$\begin{cases} -\mu\Delta\mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega \\ \mathbf{u} = \mathbf{b} & \text{on } \Gamma \end{cases} \quad (22)$$

where $\mathbf{f} = 0$. In this section, we present an extension of the embedded boundary integral method [9] to solve the *inhomogeneous* case of (22) with general \mathbf{f} .

We assume that \mathbf{f} is defined on a rectangular domain Ω' which contains Ω . Even if \mathbf{f} is defined only on Ω , it is typically possible to extend \mathbf{f} onto Ω' . The domain Ω' is discretized by a uniform grid with spacing h . The outputs of our algorithm are the velocity and pressure fields at the grid points inside domain Ω (see Figure 5). The method can be extended to adaptively refined Cartesian grids.

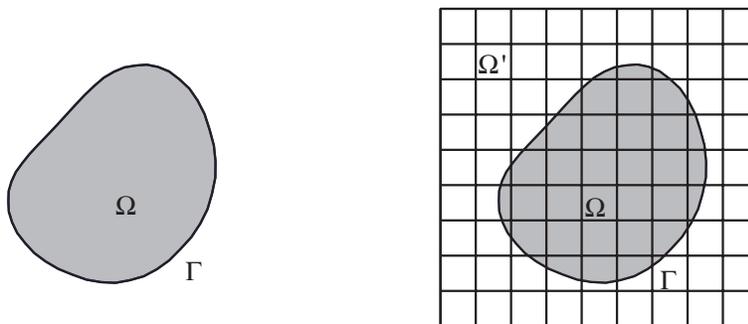


Figure 5: Domains Ω and Ω' .

The embedded boundary integral method splits the solution of the problem into several steps. First, choose a function η defined on Ω' which satisfies two conditions:

- $\eta(\mathbf{x}) = 1$ for all $\mathbf{x} \in \Omega$,
- η and all its derivatives vanish at the boundary of Ω' .

Then we replace \mathbf{f} with $\mathbf{f}\eta$ which can be extended to a smooth periodic function on \mathbb{R}^3 , with Ω' as a periodic domain. Since $\mathbf{f} = \mathbf{f}\eta$ on Ω , this replacement does not affect our original problem (22). We then decompose $\mathbf{f}\eta$ into two parts: $\mathbf{f}\eta = \mathbf{f}_c + \mathbf{f}_0$, where \mathbf{f}_c is constant over Ω' and $\int_{\Omega'} \mathbf{f}_0(\mathbf{x}) \, d\mathbf{x} = 0$. This decomposition is unique and both functions still can be extended to smooth periodic with Ω' as the periodic domain.

We decompose (22) into three problems. The first problem is defined on simpler domain Ω' with periodic boundary conditions:

$$\begin{cases} -\mu\Delta\mathbf{u}_0 + \nabla p_0 = \mathbf{f}_0 & \text{in } \Omega', \\ \operatorname{div} \mathbf{u}_0 = 0 & \text{in } \Omega'. \end{cases} \quad (23)$$

The problem has a unique solution with p_0 determined up to a constant since $\int_{\Omega'} \mathbf{f}_0(\mathbf{x}) \, d\mathbf{x} = 0$. We discretize the domain using a uniform grid with spacing h . \mathbf{u}_0 and p_0 are solved with the spectral element method based on trigonometric basis and FFT.

The second problem is

$$\begin{cases} -\mu\Delta\mathbf{u}_1 + \nabla p_1 = \mathbf{f}_c & \text{in } \Omega', \\ \operatorname{div} \mathbf{u}_1 = 0 & \text{in } \Omega', \end{cases} \quad (24)$$

with no conditions at the boundary of Ω' . We simply choose $\mathbf{u}_1 = 0$ and $p_1 = \mathbf{f}_c \cdot \mathbf{x}$ as the solution.

The last problem is a homogeneous problem over Ω :

$$\begin{cases} -\mu\Delta\mathbf{u}_2 + \nabla p_2 = 0 & \text{in } \Omega, \\ \operatorname{div} \mathbf{u}_2 = 0 & \text{in } \Omega, \\ \mathbf{u}_2 = \mathbf{b} - \mathbf{u}_0 - \mathbf{u}_1 & \text{on } \Gamma. \end{cases} \quad (25)$$

We use our boundary integral solver for this problem and evaluate \mathbf{u}_2 and p_2 for the grid points inside Ω . The solution of (23) yields values of \mathbf{u}_0 only for a regular grid in Ω' , while we need the values of \mathbf{u}_0 at the Nyström points on the boundary surface Γ for the boundary conditions of (25). We use a standard high-order spline interpolation method to obtain these values.

By linearity, velocity $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}_1 + \mathbf{u}_2$ and pressure $p = p_0 + p_1 + p_2$ evaluated at grid points inside Ω solve the original problem (22). The velocity field \mathbf{u}_2 and pressure field p_2 in Ω are evaluated as explained in Section 4.

Remark 5.1. The proposed approach works for the Laplace equation and the Navier equation. The only difference is the explicit solution of the second problem. For the Laplace equation, the second problem is

$$-\Delta u_1 = f_c \quad \text{in } \Omega'.$$

We use the explicit solution

$$u_1 = -\frac{f_c}{6}(x^2 + y^2 + z^2).$$

For the Navier equation, the second problem is

$$-\mu\Delta\mathbf{u}_1 - \frac{\mu}{1-2\nu}\nabla\operatorname{div}\mathbf{u}_1 = \mathbf{f}_c \quad \text{in } \Omega'.$$

Assuming $\mathbf{f}_c = (f_c^x, f_c^y, f_c^z)^t$, we use the explicit solution

$$\mathbf{u}_1 = -\frac{1}{4\mu} \begin{pmatrix} f_c^x(y^2 + z^2) \\ f_c^y(x^2 + z^2) \\ f_c^z(x^2 + y^2) \end{pmatrix}.$$

6 Numerical Results

Implementation. The algorithms described in the previous sections have been implemented in C++. External libraries include PETSc [6] and FFTW [17]. All tests were performed on a Linux PC with 1GB memory and a 2GHz CPU. The boundary of the domains of the test examples were represented either using analytic parameterization or using the method of [48]. We set the accuracy of the kernel-independent FMM to be 10^{-8} so that the error introduced by FMM is small compared to the typical quadrature error.

Special solution tests. We use closed form solutions of the boundary integral equations to test the accuracy and efficiency of the singular and nearly-singular quadrature rules described in the previous sections.

Table 1 shows the results for the Laplace equation on a bounded domain. The boundary surface is shown in Figure 6(c). The special solution we choose is $u(\mathbf{x}) = 1$ for $\mathbf{x} \in \Omega$. In this case the exact double layer density is equal to 1 as well. To test the singular quadrature algorithm, we choose N points on the boundary surface, and compare the exact solution $u = 1$ and the numerical solution at these points computed using the exact double layer density. To test the nearly-singular quadrature algorithm, we choose N points from the interior the domain which are very close to the surface.

h	N	T_s	$Error_s$	$Ratio_s$	T_n	$Error_n$	$Ratio_n$
0.1000	1688	1.660e+00	7.551e-04	1.027e+01	5.100e-01	5.259e-04	7.203e+00
0.0500	5480	8.980e+00	7.352e-05	1.348e+01	2.950e+00	7.301e-05	1.208e+01
0.0250	20736	6.928e+01	5.456e-06	7.406e+00	2.229e+01	6.046e-06	7.683e+00
0.0125	78792	4.636e+02	7.367e-07		1.039e+02	7.869e-07	

Table 1: Singular and Nearly-singular integration of the Laplace kernel. Columns: h is the spacing of discretization, N is the number of Nyström points, T_s is the time spent on singular evaluation, $Error_s$ is the error of singular evaluation, $Ratio_s$ is the ratio between the errors of two consecutive singular evaluations, T_n is the time spent on nearly-singular evaluation, $Error_n$ is the error of nearly-singular evaluation, and finally, $Ratio_n$ the ratio between the errors of two consecutive nearly-singular evaluations.

For the Stokes equations, we use the unit ball at the test domain. The special solution we use is the rigid body fluid rotation. The associated velocity field is $\mathbf{u}(\mathbf{x}) = \boldsymbol{\omega} \times \mathbf{x}$ for $\mathbf{x} \in \Omega$ where $\boldsymbol{\omega}$ is the angular velocity vector of the rotation and has unitary norm. In this case the double layer density $\phi(\mathbf{x})$ is equal to $\boldsymbol{\omega} \times \mathbf{x}$ as well. Table 2 summarizes the test results for singular and nearly-singular quadrature methods for the double layer Stokes kernel. For both Stokes and Laplace equations, numerical results confirm higher-order convergence of our quadrature rules.

h	N	T_s	$Error_s$	$Ratio_s$	T_n	$Error_n$	$Ratio_n$
0.1000	2400	2.410e+00	3.489e-04	7.977e+00	4.460e+00	5.072e-04	9.916e+00
0.0500	9600	1.621e+01	4.374e-05	1.575e+01	2.113e+01	5.115e-05	1.614e+01
0.0250	38400	1.228e+02	2.778e-06	1.156e+01	1.664e+02	3.169e-06	1.319e+01
0.0125	149784	9.142e+02	2.403e-07		1.333e+03	2.403e-07	

Table 2: Singular and Nearly-singular integration of the Stokes kernel. The headers for the columns are the same as in Table 1.

Homogeneous and inhomogeneous boundary value problems. Next we show results for several homogeneous and inhomogeneous elliptic boundary value problems. We use PETSc’s GMRES solver to solve linear systems with restart number 40 with halting relative tolerance equal to 10^{-9} . In every test, we embed the domain into a dense regular grid, the mesh size of which is equal to the spacing h used to discretize the boundary of the domain. As a result, the distances between many Cartesian gridpoints and the domain

boundary are less than h , which allows us to fully test the nearly-singular evaluation algorithm. The solution is evaluated at the Cartesian gridpoints inside the domain. In every test, this evaluation step is 2-3 times slower than *one* iteration of the boundary integral solver. We report the relative error which is the ratio between the L^2 norms of the absolute error and the exact solution.

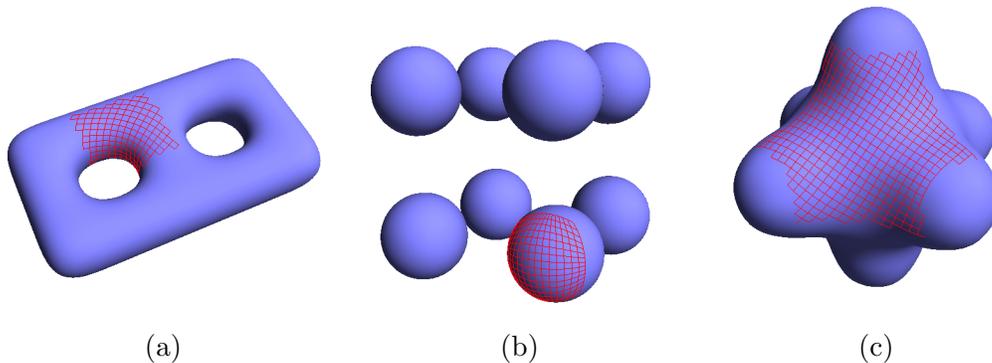


Figure 6: Boundary surfaces of the test domains. For each plot, the grid shows the Nyström points in a single patch.

Example 1. In this example, we solve the Laplace equation on a single-surface bounded domain with non-trivial topological structure. The boundary surface is shown in Figure 6(a). It is embedded in the cube $[-1, 1]^3$. The exact solution we use is $u(x, y, z) = 1 + xyz$. The results are shown in Table 3.

h	N	Iters.	FMM(s)	Local(s)	Total(s)	Error	Ratio
0.1000	2158	11	2.698e-01	1.565e+00	2.019e+01	3.494e-03	2.563e+01
0.0500	7390	10	1.452e+00	1.023e+01	1.168e+02	1.363e-04	1.566e+01
0.0250	26956	8	5.592e+00	8.501e+01	7.248e+02	8.704e-06	5.558e+00
0.0125	103209	8	2.027e+01	5.367e+02	4.456e+03	1.566e-06	

Table 3: Example 1. Laplace equation on a bounded domain with boundary shown in Figure 6(a). Columns: the discretization spacing h , the number of quadrature points N , the number of GMRES iterations, the number of seconds for each FMM evaluation, the number of seconds for local corrections for singularities, the total solution time, the relative L^2 error of the solution at the gridpoints, and the ratio between the errors of two consecutive discretizations.

Example 2. In this example, we solve the Stokes equations on an unbounded domain. The boundary surface consists of eight spheres in the cube $[-1, 1]^3$ (see Figure 6(b)). The exact solution we choose is

$$\mathbf{u}(\mathbf{x}) = \sum_{m=1}^M (S(\mathbf{x} - \mathbf{p}_m)\alpha_m + R(\mathbf{x} - \mathbf{p}_m)\beta_m)$$

where S is the Stokeslet, R is the rotlet, α_m and β_m are unit vectors with random direction, and \mathbf{p}_m are points in the spheres different from the points \mathbf{z}_m used in the integral formulation. The test results are shown in Table 4.

h	N	Iters.	FMM(s)	Local(s)	Total(s)	Error	Ratio
0.1000	3072	22	2.050e+00	1.070e+00	6.864e+01	1.408e-03	9.337e+00
0.0500	12288	21	1.080e+01	8.250e+00	4.001e+02	1.508e-04	5.054e+01
0.0250	43200	23	6.927e+01	6.566e+01	3.103e+03	2.984e-06	

Table 4: Example 2. Stokes equations on an unbounded domain with boundary shown in Figure 6(b).

Example 3. We solve the Navier equation on a single-surface bounded domain. The boundary surface (Figure 6(c)) is contained in $[-1, 1]^3$. The exact solution we choose is the sum of eight Green functions of the Navier equation centered at points $(\pm 1, \pm 1, \pm 1)$. The force vector for each Green function is a random unit vector. The results are shown in Table 5.

h	N	Iters.	FMM(s)	Local(s)	Total(s)	Error	Ratio
0.1000	2320	13	1.170e+00	1.840e+00	3.913e+01	9.822e-04	2.860e+01
0.0500	8504	12	7.920e+00	1.234e+01	2.431e+02	3.435e-05	1.087e+01
0.0250	31528	12	4.337e+01	1.123e+02	1.868e+03	3.160e-06	

Table 5: Example 3. Navier equation on a single-surface bounded domain with boundary shown in Figure 6(c).

Example 4. In this example we test the embedded boundary integral method described in Section 5 on an inhomogeneous Laplace problem

$$-\Delta u = f$$

on an single-surface bounded domain. The boundary surface (Figure 6(c)) is contained in $[-0.8, 0.8]^3$. We use the exact solution

$$\begin{cases} u(x, y, z) = \exp(\sqrt{2}\pi x) \sin(\pi(y + z)) + \frac{1}{6}(x^3 + y^3 + z^3) \\ f(x, y, z) = -(x + y + z) \end{cases}.$$

We embed the domain into the regular domain $[-1, 1]^3$. The FFT-based uniform grid solver takes only about a couple of seconds in all the runs. The performance of the boundary integral solver component, along with the overall error, is shown in Table 6.

Example 5. This example demonstrates handling of more complex geometry. We solve the Laplace problem on an unbounded domain with boundary shown in Figure 7(a), contained in $[-1, 1]^3$. The solution is set to be 1 on the boundary and approaches 0 at infinity. The performance is shown in Table 7. Figure 7(b) shows the potential field on several slices of the domain.

h	N	Iters.	FMM(s)	Local(s)	Total(s)	Error	Ratio
0.1000	1688	10	1.700e-01	1.250e+00	1.420e+01	3.166e-03	1.295e+01
0.0500	5480	10	1.140e+00	7.460e+00	8.600e+01	2.445e-04	1.619e+01
0.0250	20736	10	5.120e+00	6.346e+01	6.858e+02	1.510e-05	7.565e+00
0.0125	78792	10	1.883e+01	4.554e+02	4.742e+03	1.996e-06	

Table 6: Example 4. Poisson equation on a single-surface bounded domain (Figure 6(c)). Only the data for the boundary integral solver are shown.

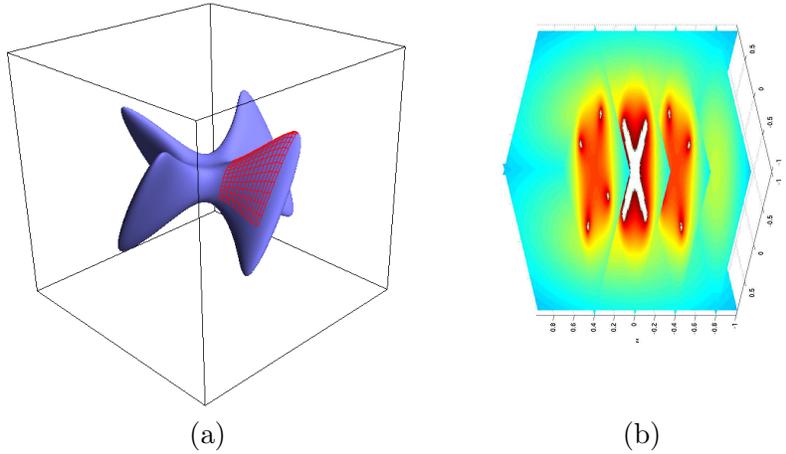


Figure 7: Example 5. Laplace equation on an unbounded domain. (a) the boundary surface of the domain and the cubic region for which the solution is evaluated. (b) cross-section view of the solution.

h	N	Iters.	FMM(s)	Local(s)	Total(s)
0.1000	1928	26	2.200e-01	1.390e+00	4.186e+01
0.0500	7720	25	1.070e+00	1.032e+01	2.848e+02
0.0250	29144	21	6.510e+00	8.822e+01	1.989e+03

Table 7: Example 5. Laplace equation on an unbounded domain (Figure 7).

Example 6. In this example, we compute the Stokes flow around a static object (Figure 8(a)). The flow velocity is $(0, 0, 1)$ at infinity and zero at the surface of the object. We evaluate the solution on a regular grid in box $(\pm 1, \pm 1, \pm 2)$. Figure 8(b) shows the magnitude of the flow velocity for several slices of the domain and the streamlines of the flow. The performance of the solver is reported in Table 8.

h	N	Iters.	FMM(s)	Local(s)	Total(s)
0.0500	5465	84	3.750e+00	6.600e+00	8.694e+02
0.0250	19684	69	2.031e+01	5.545e+01	5.227e+03

Table 8: Example 6. Stokes equations on an unbounded domain. (Figure 8).

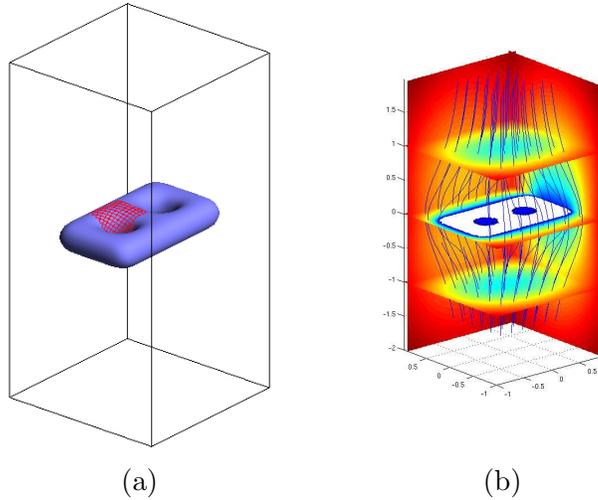


Figure 8: Example 6. Stokes equations on an unbounded domain. (a) the boundary surface of the domain and the rectangle region where the solution is evaluated. (b) velocity magnitude for several cross-sections and streamlines of the fluid field.

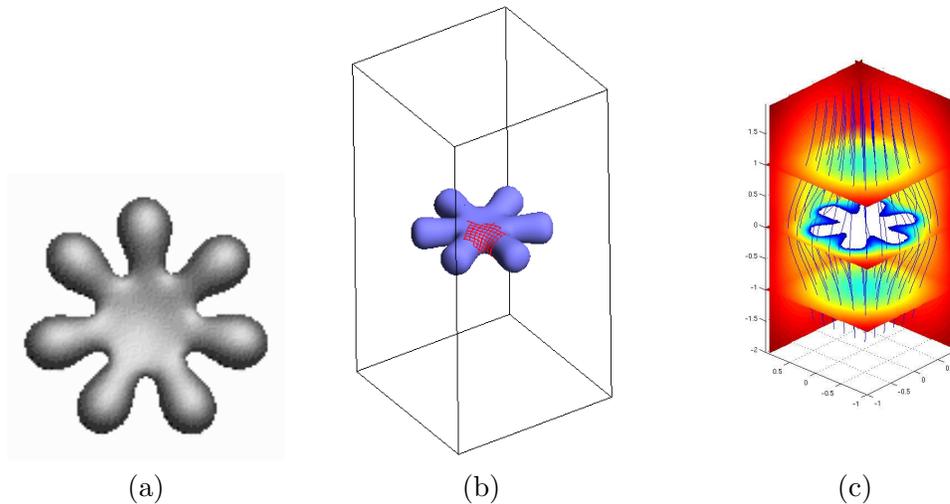


Figure 9: Example 7. Stokes equations with an unbounded domain. (a) A starfish vesicle from [45]. (b) The boundary surface of the domain. (c) The velocity magnitude and flow streamlines.

Example 7. In this example, we compute the Stokes flow around a shape similar to starfish vesicles [45], Figure 9(a). The boundary of the domain is shown in Figure 9(b). The flow velocity is $(0, 0, -1)$ at infinity and zero at the surface of the object. We evaluate the solution in a regular grid in box $(\pm 1, \pm 1, \pm 2)$. Figure 9(c) shows the magnitude of the flow velocity on several slices of the domain and the streamlines of the flow. The performance of the solver is reported in Table 9.

In the first four examples for which the exact solutions are known, we observe high-order convergence of our algorithms. In general, the error decreases by a factor of 8 to 30 each time we half the discretization spacing h .

h	N	Iters.	FMM(s)	Local(s)	Total(s)
0.0500	4304	120	2.955e+00	5.745e+00	1.044e+03
0.0250	16636	91	1.823e+01	5.031e+01	6.237e+03

Table 9: Example 7. Stokes equations with an unbounded domain.

In all examples, we observe that the FMM time increases linearly with respect to the number of the quadrature points. On the other hand, the local correction for integrating singularities grows more rapidly: it increases by a factor of 8 as the number of Nyström points quadruples. These results match the analysis in Section 3.2.

7 Conclusions

We have presented a high-order 3D boundary integral solver for elliptic PDEs for arbitrary domains with smooth boundaries. For problems requiring evaluation at sufficiently many points in the spatial domain the solver has optimal complexity (linear in the number of evaluation points).

Numerical studies confirmed the expected asymptotic behavior for a variety of geometries and equations. Our boundary integral solver was also used to solve inhomogeneous problems on arbitrary 3D domains with smooth boundaries.

In this paper we have not considered the problem of parallelizing this type of solvers. While the two most critical components (the kernel-independent FMM and FFTW) have been parallelized [47], other components (local corrections for singular integration, nearly singular integration) are not. Although they should be relatively easy to parallelize due to local nature of computations involved, a complete scalable implementation is far from trivial and is a topic for future work.

While exhibiting expected asymptotic behavior, our current implementation is far from optimal as the emphasis was on correctness rather than efficiency. Performance of many parts of the code can be improved.

Finally, domains with sharp features play an important role in many applications. Considering extensions of this approach to piecewise smooth domains is another important research direction.

Acknowledgments. This research was supported by Sloan Foundation Fellowship, NYU Dean’s Dissertation Fellowship, and NSF DMS-9980069 and DOE DE-FG02-04ER25646 awards. The authors would like to thank Leslie Greengard and Michael Shelley for numerous discussions and suggestions.

A Error Estimates for Quadrature Rules

In this appendix we derive the error bounds for the quadrature rules described in Section 3. These rules are used to compute the integrals

$$\int_U D(\mathbf{x}, g(c))\psi(c) dc \quad \text{and} \quad \int_U K(\mathbf{x}, g(c))(\psi(c) - \psi(c')) dc \quad (26)$$

over planar domains U for $\mathbf{x} = g(c')$ for a fixed $c' \in U$. We assume that the values at quadrature points are computed exactly, $g(\cdot)$ is C^∞ -continuous but not necessarily analytic,

its Jacobian is of maximal rank and the functions ϕ are C^M continuous for some $M \geq 3$. The assumption on continuity of $g(\cdot)$ can be relaxed if necessary, but we use the stronger assumption as all surface parameterizations we use are C^∞ -continuous.

We use $\partial_z^m f$ to denote m -th derivative of f with respect to variable z . We use C to denote constants in estimates which may be different in different formulas.

The error of one-dimensional trapezoidal rule applied to a periodic C^M -continuous function on its period $[0, t]$ is

$$\epsilon_{\text{trap}}^{\text{1D}} \leq C(M)h^M \sup_{[0,t]} |\partial^M f|. \quad (27)$$

Let I_z be the operator $f \rightarrow \int_0^{t_z} f(z, \dots) dz$, where f is periodic in z with the period equal to t_z . Let I_z^h be the one-dimensional trapezoid rule operator, with spacing $h = t_z/N$ in variable z : $I_z^h f = \sum_{i=0}^N h f(z_i, \dots)$. Consider a function of two variables u and v . Then we observe that operators applied to different variables commute. This allows us to express the error for the two-dimensional rule as follows:

$$\epsilon_{\text{trap}}^{\text{2D}} = I_u I_v f - I_u^{h_u} I_v^{h_v} f = \left(I_u I_v f - I_u^{h_u} I_v f \right) + \left(I_u^{h_u} I_v f - I_u^{h_u} I_v^{h_v} f \right).$$

The first term is the error of the trapezoidal rule used to integrate $I_v f$ with respect to u ; the second term is the error of the trapezoidal rule used to integrate $I_u^{h_u} f$ with respect to v . Applying the error estimates and observing that ∂_u commutes with I_v and ∂_v commutes with $I_u^{h_u}$, we obtain the estimate

$$\epsilon_{\text{trap}}^{\text{2D}} < C(M)h_u^M \sup_{[0,t_u]} \left| \int_0^{t_v} \partial_u^M f dv \right| + C(M)h_v^M \sup_{[0,t_v]} \left| \sum_0^{N_v} (\partial_v^M f)(u, ih_v) \cdot h_u \right|.$$

Bounding the integral and the sum from above, we obtain the *error bound for the two-dimensional trapezoidal rule*:

$$\epsilon_{\text{trap}}^{\text{2D}} < C(M) \left(h_u^M \sup_{[0,t_u]} |\partial_u^M f| + h_v^M \sup_{[0,t_v]} |\partial_v^M f| \right). \quad (28)$$

We first prove the convergence result of the quadrature rule used for $D(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})$.

Lemma A.1. *Suppose $f(q) : A \rightarrow \mathbb{R}$, is a C^M -continuous function, $f(q)|q|^{-k}$, $k < M$, is bounded, and $A \subset \mathbb{R}^2$ contains the disk of radius a centered at zero. Let $q(\rho, \theta) = (\rho \cos \theta, \rho \sin \theta)$. Then*

1. $f(q(\rho, \theta))\rho^{-k}$ is C^{M-1} -continuous on $[-a, a] \times [0, \pi]$.
2. The derivatives of $f(q)|q|^{-k}$ of order $m \leq M - 1$ are bounded from above by $C\rho^{-m}$.

Proof. let $f(\rho, \theta) = f(q(\rho, \theta))$. For $\rho \neq 0$, the first part of the lemma follows from the chain rule; we only need to show that $\partial_\rho^j \partial_\theta^i (\rho^{-k} f)$ converges as $\rho \rightarrow 0$, for $i + j \leq M - k$, and the limit is a continuous function of θ . To simplify exposition, we consider ∂_ρ^j only; the general case is analogous.

Let $T_m^f(\rho)$, $M \geq m \geq k$, be the Taylor polynomial of degree m at zero for the function $f(\rho, \theta)$ with respect to ρ , with coefficients dependent on θ . Then f can be expressed as

$$f = T_m^f + R^{m+1},$$

where $R^{m+1}\rho^{-m} \rightarrow 0$ for $\rho \rightarrow 0$. Because $f\rho^{-k}$ is bounded, the first $k-1$ terms of T_m^f are zero. Also, $\partial_\rho^j T_m^f$ is the Taylor polynomial of degree $m-j$ for $\partial_\rho^j f$:

$$\partial_\rho^j f = \partial_\rho T_m^f + R_j^{m+1-j},$$

where $R_j^{m+1-j}\rho^{-(m-j)} \rightarrow 0$ for $\rho \rightarrow 0$. By comparing the formulas for f and $\partial_\rho^j f$ we obtain the equality $R_j^{m-j+1} = \partial_\rho^j R^{m+1}$.

Consider

$$\partial_\rho^j(\rho^{-k}f) = \partial_\rho^j(\rho^{-k}T_m^f) + \partial_\rho^j(\rho^{-k}R^{m+1}).$$

Because the first $k-1$ terms of T_m^f are zero, the first term is a derivative of a polynomial and is well-defined for any ρ ; the limit at $\rho = 0$ can be easily checked to be $(j!/k!)(\partial_\rho^{k+j}f)(0)$. It remains to verify that the limit of $\partial_\rho^j(\rho^{-k}R^{m+1})$. By the product derivative formula,

$$\partial_\rho^j(\rho^{-k}R^{m+1}) = \sum \binom{l}{j} \partial_\rho^l R^{m+1} \frac{k!}{(j-l)!} \rho^{-k-j+l}.$$

As $\partial_\rho^l R^{m+1} = R_l^{m+1-l}$, each term can be rewritten as $C(R_l^{m-l+1}\rho^{-(m-l)})\rho^{m-k-j}$. As $R_l^{m+1-l}\rho^{-(m-l)} \rightarrow 0$ as $\rho \rightarrow 0$ (it is a remainder term in Taylor expansion of $\partial_\rho^l f$ of degree $m-l$), the whole sum converges to zero if $m \geq j+k$. As we can choose m in the range from 0 to M , we conclude that the limit of $\partial_\rho^j(\rho^{-k}f)$ at $\rho = 0$ is $(j!/k!)(\partial_\rho^{k+j}f)(0)$, if $k+j \leq M$. Its continuity follows from the continuity of derivatives of f .

The second part of the lemma easily follows from the first. If f is differentiable in polar coordinates, then at any point away from zero, we can express derivatives with respect to q_1 and q_2 as $\partial_1 = \cos\theta\partial_\rho - \rho^{-1}\sin\theta\partial_\theta$ and $\partial_2 = \sin\theta\partial_\rho + \rho^{-1}\cos\theta\partial_\theta$ respectively. As $f\rho^{-1}$ is bounded and all derivatives up to order $M-1$ with respect to θ and ρ are bounded, in the expressions for the derivatives of f the only factors that are not bounded have the form ρ^{-l} , and one can show by induction that for derivatives of order m , the maximal value of l is m . \square

We observe that if the function f is C^∞ -continuous, the lemma can be applied for arbitrary M .

Lemma A.2. *Let $g(c)$ be a C^∞ -continuous regular parametrization of a part of the boundary Γ , c' a fixed point in the parametrization domain C containing a disk of radius a centered at c' , and (ρ, θ) is a polar coordinate system centered at c' . The kernel $D(\mathbf{x}, \mathbf{y})$ defined by (2) has the following properties on the boundary:*

1. $\rho D(g(c'), g(c(\rho, \theta)))$ is a C^∞ -continuous function of ρ and θ on $[-a, a] \times [0, \pi]$;
2. as $\rho \rightarrow 0$, the m -th order derivatives of $D(g(c'), g(c))$ with respect to c are $O(\rho^{-(m+1)})$.

Proof. Let $\mathbf{r} = g(c') - g(c)$. The matrix function ρD can be decomposed into a combination of several functions, each satisfying the conditions of Lemma A.1:

$$\rho D(g(c'), g(c)) = -\frac{3}{4\pi} (\mathbf{r}/\rho \otimes \mathbf{r}/\rho) \frac{(\mathbf{r} \cdot \mathbf{n}(g(c)))}{\rho^2} (\mathbf{r}^2/\rho^2)^{-5/2}.$$

As g is nonsingular at $c = c'$, all components of \mathbf{r}/ρ are bounded from below and so is \mathbf{r}^2/ρ^2 . For $(\mathbf{r} \cdot \mathbf{n}(g(c)))$, an explicit calculation using Taylor expansions shows that the linear terms in ρ vanish, and the first nonzero terms are of order ρ^2 .

By Lemma A.1, \mathbf{r}/ρ , \mathbf{r}^2/ρ^2 and $(\mathbf{r} \cdot \mathbf{n}(g(c(\rho, \theta))))$ are all C^∞ -continuous in polar coordinates.

As r^2/ρ^2 can be bounded from below by a constant $C > 0$ the negative power function in the last factor of the decomposition does not affect smoothness, so $\rho D(g(c'), g(c(\rho, \theta)))$ is C^∞ -continuous.

Using the second part of Lemma A.1, we conclude that the growth of m -th derivatives of ρD is bounded by ρ^m . To obtain the rate of growth for derivatives of D we apply the product derivative formula to $(\rho D)\rho^{-1}$; the resulting rate is ρ^{m+1} . \square

Theorem A.3. *The error of the quadrature rules of Section 3.1 for the integral $(D\phi)(\mathbf{x})$, evaluated at any \mathbf{x} on the boundary Γ is $O(h^{\frac{M-1}{2}})$ if ϕ is C^M continuous.*

Proof. We consider two parts of the integral separately.

Nonsingular part. In this case the singularity is eliminated by multiplying D by $(1 - \eta_0(c))$, with $\eta_0(c) = \chi(c/\sqrt{h})$ where $\chi : [0, \infty) \rightarrow [0, 1]$ is a C^∞ function which is 1 in a neighborhood of zero, and $\chi(r) = 0$ for $r \geq 1$.

To apply the estimate for the trapezoidal rule (28), we need to estimate the magnitude of the derivative of the integrand which depends on h . By Lemma A.2 m -th derivatives of D are $O(\rho^{-m-1})$; as $1 - \eta_0(c)$ vanishes in a circle of radius $\sqrt{h}/4$, derivatives of D contribute to the derivatives of the product only outside this circle, and can be bounded by $Ch^{-(m+1)/2}$.

We obtain the bound $h^{m/2}$ for the derivatives of $(1 - \eta_0(c))$ of order m by applying the chain rule to $\chi(\cdot/\sqrt{h})$.

Substituting the bounds for $\partial_j^m D$ and $\partial_j^m (1 - \eta_0(c))$ into the product derivative formula for $\partial_j^M (D(g(c'), g(c))(1 - \eta(c))\phi(c))$, we obtain

$$|\partial_j^M (D(g(c'), g(c))(1 - \eta(c))\phi(c))| < Ch^{-\frac{M+1}{2}}.$$

Combining this estimate with the 2D trapezoidal rule error estimate we obtain the error bound for the first part of the integral: $O(h^{M-(M+1)/2}) = O(h^{(M-1)/2})$.

Singular part. This part is integrated using the trapezoidal rule in polar coordinates. We integrate $\rho D(g(c'), g(c(\rho, \theta)))\phi(\rho, \theta)\eta_0(\rho)$; We denote $f(\rho, \theta) = \rho D(g(c'), g(c(\rho, \theta)))\phi(\rho, \theta)$. By Lemma A.1, f is a C^M -continuous function. The function $f\eta_0$ vanishes for $\rho = 1/\sqrt{h}$ and is periodic in θ ; therefore, it can be extended to a periodic C^M -continuous function on the plane and the trapezoidal rule error estimate applies.

The derivatives $\partial_\theta^m (f\eta_0)$ are all bounded with respect to h , as η_0 does not depend on θ . Similarly to the nonsingular case, we obtain estimates for $\partial_\rho^m (f\eta_0)$ by using the chain rule to bound derivatives of η_0 from above by $Ch^{-m/2}$, and use the product derivative rule and boundedness of derivatives of f to extend the bound to $\partial_\rho^m (f\eta_0)$.

Applying (28), with $h_\theta = \sqrt{h}$, $h_\rho = h$, and with $\partial_\theta^M (f\eta_0) = O(1)$ and $\partial_\rho^M (f\eta_0) = O(h^{-M/2})$, we obtain the error bound of order $O(h^{M/2})$.

Combining the error bounds for the singular and nonsingular parts yields the statement of the theorem. \square

We then prove the convergence result of the quadrature rule used for $K(\mathbf{x}, \mathbf{y})(\phi(\mathbf{y}) - \phi(\mathbf{x}))$.

Lemma A.4. *Suppose that $f : \mathbb{R} \rightarrow \mathbb{R}$ can be written in the following form:*

$$f(\rho) = \frac{1}{\rho} S(\rho),$$

where S is a C^M function. Then the trapezoidal rule with quadrature points at $(k - \frac{1}{2})h$ results an $O(h^{\frac{M-1}{2}})$ error for the integral

$$\int_{-\sqrt{h}}^{\sqrt{h}} f(\rho) \chi\left(\frac{|\rho|}{\sqrt{h}}\right) d\rho.$$

Proof. We use the singularity subtraction to rearrange the integral:

$$\begin{aligned} \int_{-\sqrt{h}}^{\sqrt{h}} \frac{1}{\rho} S(\rho) \chi\left(\frac{|\rho|}{\sqrt{h}}\right) d\rho &= \int_{-\sqrt{h}}^{\sqrt{h}} \frac{1}{\rho} (S(\rho) - S(0)) \chi\left(\frac{|\rho|}{\sqrt{h}}\right) d\rho + \\ &\quad \left(\int_{-\sqrt{h}}^{\sqrt{h}} \frac{1}{\rho} \chi\left(\frac{|\rho|}{\sqrt{h}}\right) d\rho \right) S(0) \end{aligned}$$

The integrand of the first integral is C^{M-1} -continuous and can be extended periodically. The trapezoidal rule with quadrature points $(k - \frac{1}{2})h$ has $O(h^{\frac{M-1}{2}})$ accuracy for this integral. The second integral is a Cauchy integral with the principal value zero. The trapezoidal rule for this integral with $(k - \frac{1}{2})h$ as quadrature points produces zero as well, since the quadrature points, the function $\chi(\frac{|\cdot|}{\sqrt{h}})$ and the integration domain are symmetric around the origin.

Therefore, the trapezoidal rule has an overall error of order $O(h^{\frac{M-1}{2}})$ for the principal value of the integral. \square

Lemma A.5. *Let $g(c)$ be a C^∞ -continuous regular parametrization of a part of the boundary Γ , c' a fixed point in the parametrization domain C containing a disk of radius a centered at c' , and (ρ, θ) is a polar coordinate system centered at c' . Suppose ϕ is C^M .*

The function $K(\mathbf{x}, \mathbf{y})(\phi(\mathbf{y}) - \phi(\mathbf{x}))$ has the following properties on the boundary:

1. $\rho K(g(c'), g(c(\rho, \theta)))(\phi(\mathbf{y}) - \phi(\mathbf{x}))$ can be written as $S(\rho, \theta)/\rho$ in polar coordinates, where S is a C^{M-1} function.
2. As $\rho \rightarrow 0$, all m th derivatives of $K(g(c'), g(c(\rho, \theta)))(\phi(\mathbf{y}) - \phi(\mathbf{x}))$ with respect to c are $O(\rho^{-(m+3)})$ for any $m \leq M$.

Proof. We consider the function $\rho^2 K(g(c'), g(c(\rho, \theta)))(\phi(\mathbf{y}) - \phi(\mathbf{x}))$. Let $r = g(c') - g(c(\rho, \theta))$. Since

$$K(g(c'), g(c(\rho, \theta))) = \left(\frac{\mathbf{n}(\mathbf{y})}{|\mathbf{r}|^3} - 3 \frac{(\mathbf{r} \cdot \mathbf{n}(\mathbf{y}))\mathbf{r}}{|\mathbf{r}|^5} \right),$$

By Lemma A.1 implies that $\rho^3 K(g(c'), g(c(\rho, \theta)))$ is a C^∞ function in polar coordinates. Also by Lemma A.1 for C^M functions, $(\phi(\mathbf{y}) - \phi(\mathbf{x}))/\rho$ is a C^{M-1} function in the polar coordinates. Thus, $\rho^2 K(g(c'), g(c(\rho, \theta)))(\phi(\mathbf{y}) - \phi(\mathbf{x}))$ is a C^{M-1} in polar coordinates, which proves the first property.

Since $\rho^3 K(g(c'), g(c(\rho, \theta)))$ is a C^∞ function in polar coordinates, it follows that

$$\rho^3 K(g(c'), g(c(\rho, \theta)))(\phi(\mathbf{y}) - \phi(\mathbf{x}))$$

is a C^M function in polar coordinates. Using the second part of Lemma A.1 and the product derivative formula we conclude that the m th derivatives of $K(g(c'), g(c(\rho, \theta)))(\phi(\mathbf{y}) - \phi(\mathbf{x}))$ are $O(\rho^{-(m+3)})$. \square

Theorem A.6. *The error of the quadrature rule of Section 3.1 for the integral of $K(\mathbf{x}, \mathbf{y})(\phi(\mathbf{y}) - \phi(\mathbf{x}))$ evaluated at any \mathbf{x} on the boundary of Γ is $O(h^{\frac{M-3}{2}})$ if ϕ is C^M .*

Proof. The structure of the proof is similar to the proof of Theorem A.3. The integral over each patch U is decomposed into two parts:

$$\int_U K(g(c'), g(c))(1 - \eta_{c'}(c))(\psi(c) - \psi(c')) dc + \int_U K(g(c'), g(c))\eta_{c'}(c)(\psi(c) - \psi(c')) dc.$$

The integrand of the first integral is not singular. The M -th derivatives of $1 - \eta_{c'}(c)$ are $O(h^{\frac{M}{2}})$ and the M derivatives of $K(g(c'), g(c))(\psi(c) - \psi(c'))$ on the support of $1 - \eta_{c'}(c)$ can be bounded by $O(h^{\frac{M+3}{2}})$ by the second part of Lemma A.5. It follows from product derivative formula that the m -th derivatives of $K(g(c'), g(c))(1 - \eta_{c'}(c))(\psi(c) - \psi(c'))$ are of order $O(h^{\frac{M+3}{2}})$. Applying the error estimate of the 2D trapezoidal rule results an error bound of $O(h^{\frac{M-3}{2}})$.

We write the second integral, which is singular, in polar coordinates:

$$\int \int \rho K(g(c'), g(c))\eta_{c'}(c)(\psi(c) - \psi(c')) d\rho d\theta.$$

It follows from the first part of Lemma A.5 that the integrand can be written as $S(\rho, \theta)\rho$ where $S(\rho, \theta)$ is a C^{M-1} function. We use Lemma A.4 to conclude that the 1D quadrature rule for

$$\int \rho K(g(c'), g(c))\eta_{c'}(c)(\psi(c) - \psi(c')) d\rho \quad (29)$$

has an error bound $O(h^{\frac{M-2}{2}})$. Since (29) as a function of θ is periodic for $\theta \in [0, \pi]$, the convergence rate of the quadrature rule for the second integral is also $O(h^{\frac{M-2}{2}})$.

Combining the error bounded for the non-singular and singular parts, we conclude the overall error bound is $O(h^{\frac{M-3}{2}})$. □

B Derivation of Pressure Jump

We choose a local orthonormal frame $\boldsymbol{\alpha}, \boldsymbol{\beta}$ in the tangent plane at \mathbf{x} . There are three conditions on the jumps:

$$[[\mathbf{u}]] = \boldsymbol{\phi}, \quad (30)$$

$$[[-pI + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^t)]\mathbf{n}] = 0 \quad (\text{Lyapovov-Tauber condition}), \quad (31)$$

$$[[\text{div } \mathbf{u}]] = 0. \quad (32)$$

Let $\boldsymbol{\alpha}, \boldsymbol{\beta}$ be a local orthonormal frame in the tangent plane at \mathbf{x} . Differentiating in directions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (30), we get

$$[[\nabla \mathbf{u}]]\boldsymbol{\alpha} = \boldsymbol{\phi}_\alpha \quad [[\nabla \mathbf{u}]]\boldsymbol{\beta} = \boldsymbol{\phi}_\beta.$$

This implies the following six equalities:

$$\begin{aligned} \boldsymbol{\alpha}^t [[\nabla \mathbf{u}]]\boldsymbol{\alpha} &= \boldsymbol{\alpha}^t \boldsymbol{\phi}_\alpha & \boldsymbol{\alpha}^t [[\nabla \mathbf{u}]]\boldsymbol{\beta} &= \boldsymbol{\alpha}^t \boldsymbol{\phi}_\beta, \\ \boldsymbol{\beta}^t [[\nabla \mathbf{u}]]\boldsymbol{\alpha} &= \boldsymbol{\beta}^t \boldsymbol{\phi}_\alpha & \boldsymbol{\beta}^t [[\nabla \mathbf{u}]]\boldsymbol{\beta} &= \boldsymbol{\beta}^t \boldsymbol{\phi}_\beta, \\ \mathbf{n}^t [[\nabla \mathbf{u}]]\boldsymbol{\alpha} &= \mathbf{n}^t \boldsymbol{\phi}_\alpha & \mathbf{n}^t [[\nabla \mathbf{u}]]\boldsymbol{\beta} &= \mathbf{n}^t \boldsymbol{\phi}_\beta. \end{aligned}$$

Taking dot products of (31) with $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ results in

$$\boldsymbol{\alpha}^t ([[\nabla \mathbf{u}]]) + [[\nabla \mathbf{u}]]^t \mathbf{n} = 0, \quad \boldsymbol{\beta}^t ([[\nabla \mathbf{u}]]) + [[\nabla \mathbf{u}]]^t \mathbf{n} = 0.$$

Since $\operatorname{div} \mathbf{u} = \operatorname{tr}(\nabla \mathbf{u})$ and $\operatorname{tr}(A) = \operatorname{tr}(ABB^{-1}) = \operatorname{tr}(B^{-1}AB)$,

$$\operatorname{tr}((\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{n})^t [[\nabla \mathbf{u}]]) (\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{n}) = 0,$$

which is equivalent to

$$\boldsymbol{\alpha}^t [[\nabla \mathbf{u}]]) \boldsymbol{\alpha} + \boldsymbol{\beta}^t [[\nabla \mathbf{u}]]) \boldsymbol{\beta} + \mathbf{n}^t [[\nabla \mathbf{u}]]) \mathbf{n} = 0.$$

Combining all conditions on $[[\nabla \mathbf{u}]])$, we obtain

$$[[\nabla \mathbf{u}]]) = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{n}) \begin{pmatrix} \boldsymbol{\alpha}^t \phi_\alpha & \boldsymbol{\alpha}^t \phi_\beta & -\mathbf{n}^t \phi_\alpha \\ \boldsymbol{\beta}^t \phi_\alpha & \boldsymbol{\beta}^t \phi_\beta & -\mathbf{n}^t \phi_\beta \\ \mathbf{n}^t \phi_\alpha & \mathbf{n}^t \phi_\beta & -\boldsymbol{\alpha}^t \phi_\alpha - \boldsymbol{\beta}^t \phi_\beta \end{pmatrix} (\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{n})^t.$$

Putting it back into (31), we have

$$[[p]] = -2\mu(\boldsymbol{\alpha}^t \phi_\alpha + \boldsymbol{\beta}^t \phi_\beta).$$

C Error Estimate of Nearby Evaluation

In this appendix, we prove that the algorithm described in Section 4 achieves high-order convergence.

Lemma C.1. *For $\mathbf{x} \in \Omega_0$, the error of the algorithm is $O(h^{\frac{M}{2}-1})$ if ϕ is C^M -continuous.*

Proof. For a fixed \mathbf{x} , the integral over a single chart U is

$$\int_U D(\mathbf{x}, g(c)) \psi(c) \, dc$$

where $\psi(c) = w(g(c)) \phi(g(c)) J(c)$. In order to estimate the error of the trapezoidal rule, we need to estimate $\partial_j^M (D(\mathbf{x}, g(c)) \psi(c))$ for $j = 1, 2$. Since g is C^∞ and ϕ is C^M , all M th order derivatives of $\psi(c)$ is uniformly bounded and it suffices to control $\partial_j^M D(\mathbf{x}, g(c))$. Furthermore, all M th order derivatives of $D(\mathbf{x}, g(c))$ with respect to the chart parameterization is bounded by a multiple of supremum of all M th order derivatives of $D(\mathbf{x}, \mathbf{y})$ with respect to \mathbf{y} .

Since $D(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{r}|^2} S(\mathbf{r})$ for a C^∞ function S , we have the following estimate for any multiindex β with $|\beta| = M$,

$$|\partial_{\mathbf{y}}^\beta D(\mathbf{x}, \mathbf{y})| \leq C \frac{1}{|\mathbf{r}|^{M+2}}$$

where $\mathbf{r} = \mathbf{x} - \mathbf{y}$ and the constant C is independent of \mathbf{r} . Therefore, we have the following estimates for the M -th derivatives of $f(c)$:

$$|\partial_j^M (D(\mathbf{x}, g(c)) \psi(c))| \leq C \frac{1}{|\mathbf{r}|^{M+2}} \leq C \frac{1}{h^{M/2+1}} \quad (33)$$

for $j = 1, 2$. By (28), the error of using the trapezoidal rule to integrate f is bounded by

$$C \left(\sup_{j,c} |\partial_j^M (D(\mathbf{x}, g(c)) \psi(c))| \right) h^M = O(h^{\frac{M}{2}-1}).$$

□

Lemma C.2. For $\mathbf{x} \in \Omega_1$, the error of the algorithm is $O(h^{\frac{M}{2}-2})$ if ϕ is C^M -continuous.

Proof. We apply the same estimate (33), except we use h instead of \sqrt{h} , obtaining the error bound of order $O(h^{-M-2})$. Therefore, the error of using the trapezoidal rule on the refined grid with spacing $h^{3/2}$ can be bounded by

$$C \left(\sup_{j,c} |\partial_j^M (D(\mathbf{x}, g(c))\psi(c))| \right) h^{\frac{3M}{2}} = O(h^{\frac{M}{2}-2}).$$

□

Lemma C.3. For $\mathbf{x} \in \Omega_2$, the error of the algorithm of is $O(h^{\min(\frac{M}{2}-2, L)})$ if ϕ is C^M -continuous.

Proof. The limit velocity at \mathbf{x}_0 , $\frac{1}{2}\phi(\mathbf{x}_0) + D\phi(\mathbf{x}_0)$, has convergence rate $O(h^{\frac{M}{2}-1})$ by Theorem A.3. From the previous lemma, we know that the computed values of $(D\phi)$ on \mathbf{x}_l , $l = 1, \dots, L$ converge as $O(h^{\frac{M}{2}-2})$. The error introduced by the Lagrange interpolation procedure is of order $O(h^L)$. Therefore, the overall error is at most $O(h^{\min(\frac{M}{2}-2, L)})$. □

Combining these lemmas, we are ready to state the following theorem:

Theorem C.4. For any $\mathbf{x} \in \Omega$, the error of the algorithm described in Section 4 can be bounded by $O(h^{\min(\frac{M}{2}-2, L)})$ if ϕ is C^M -continuous.

References

- [1] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM J. Sci. Comput.*, 14(1):159–184, 1993.
- [2] K. Atkinson. Numerical integration on the sphere. *J. Austral. Math. Soc. Ser. B*, 23(3):332–347, 1981. 0334-2700.
- [3] K. Atkinson. Quadrature of singular integrands over surfaces. *Electron. Trans. Numer. Anal.*, 17:133–150 (electronic), 2004. 1068-9613.
- [4] K. E. Atkinson. The numerical solution of Laplace’s equation in three dimensions. *SIAM J. Numer. Anal.*, 19(2):263–274, 1982. 0036-1429.
- [5] K. E. Atkinson. *The numerical solution of integral equations of the second kind*, volume 4 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 1997.
- [6] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. PETSc 2.0 users manual. Technical Report ANL-95/11 - Revision 2.0.24, Argonne National Laboratory, 1999.
- [7] J. T. Beale and M.-C. Lai. A method for computing nearly singular integrals. *SIAM J. Numer. Anal.*, 38(6):1902–1925 (electronic), 2001. 1095-7170.
- [8] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. I. *Comm. Pure Appl. Math.*, 44(2):141–183, 1991.
- [9] G. Biros, L. Ying, and D. Zorin. A fast solver for the Stokes equations with distributed forces in complex geometries. *J. Comput. Phys.*, 193(1):317–348, 2004.

- [10] C. Brebbia, J. Telles, and L. Wrobel. *Boundary element techniques*. Springer-Verlag,, Berlin, 1984.
- [11] O. P. Bruno and L. A. Kunyansky. A fast, high-order algorithm for the solution of surface scattering problems: Basic implementation, tests, and applications. *Journal of Computational Physics*, 169:80–110, 2001.
- [12] L. F. Canino, J. J. Ottusch, M. A. Stalzer, J. L. Visser, and S. M. Wandzura. Numerical solution of the Helmholtz equation in 2D and 3D using a high-order Nyström discretization. *Journal of Computational Physics*, 146(2):627–663, 1998.
- [13] G. Chen and J. Zhou. *Boundary element methods*. Computational Mathematics and Applications. Academic Press Ltd., London, 1992.
- [14] W. Dahmen. Multiscale and wavelet methods for operator equations. *Multiscale problems and methods in numerical simulations*, 1825:31–96, 2003. Lecture Notes in Math. Springer Berlin.
- [15] W. Dahmen, B. Kleemann, S. Prössdorf, and R. Schneider. A multiscale method for the double layer potential equation on a polyhedron. In *Advances in computational mathematics (New Delhi, 1993)*, volume 4 of *Ser. Approx. Decompos.*, pages 15–57. World Sci. Publishing, River Edge, NJ, 1994.
- [16] S. Erichsen and S. A. Sauter. Efficient automatic quadrature in 3-d Galerkin BEM. *Comput. Methods Appl. Mech. Engrg.*, 157(3-4):215–224, 1998. Seventh Conference on Numerical Methods and Computational Mechanics in Science and Engineering (NMCM 96) (Miskolc) 0045-7825.
- [17] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. special issue on "Program Generation, Optimization, and Platform Adaptation".
- [18] M. Ganesh, I. G. Graham, and J. Sivaloganathan. A new spectral boundary integral collocation method for three-dimensional potential problems. *SIAM J. Numer. Anal.*, 35(2):778–805 (electronic), 1998. 1095-7170.
- [19] I. G. Graham, W. Hackbusch, and S. A. Sauter. Hybrid galerkin boundary elements: theory and implementation. *Numer. Math.*, 86(1):139–172, 2000. 0029-599X.
- [20] I. G. Graham and I. H. Sloan. Fully discrete spectral boundary integral methods for helmholtz problems on smooth closed surfaces in R^3 . *Numer. Math.*, 92(2):289–323, 2002. 0029-599X.
- [21] L. Greengard, M. C. Kropinski, and A. Mayo. Integral equation methods for Stokes flow and isotropic Elasticity in the plane. *Journal of Computational Physics*, 125:403–414, 1996.
- [22] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.
- [23] W. Hackbusch. *Integral equations*, volume 120 of *International Series of Numerical Mathematics*. Birkhäuser Verlag, Basel, 1995. Theory and numerical treatment, Translated and revised by the author from the 1989 German original.

- [24] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54(4):463–491, 1989.
- [25] N. Heuer, M. Maischak, and E. P. Stephan. Exponential convergence of the hp -version for the boundary element method on open surfaces. *Numer. Math.*, 83(4):641–666, 1999. 0029-599X.
- [26] N. Heuer, M. E. Mellado, and E. P. Stephan. A p -adaptive algorithm for the BEM with the hypersingular operator on the plane screen. *Internat. J. Numer. Methods Engrg.*, 53(1):85–104, 2002. p and hp finite element methods: mathematics and engineering practice (St. Louis, MO, 2000) 0029-5981.
- [27] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Boundary integral methods for multi-component fluids and multiphase materials. *J. Comput. Phys.*, 169(2):302–362, 2001. 0021-9991.
- [28] Q. Huang and T. A. Cruse. Some notes on singular integral techniques in boundary element analysis. *Internat. J. Numer. Methods Engrg.*, 36(15):2643–2659, 1993. 0029-5981.
- [29] P. R. Johnston. Application of sigmoidal transformations to weakly singular and near-singular boundary element integrals. *Internat. J. Numer. Methods Engrg.*, 45(10):1333–1348, 1999. 0029-5981.
- [30] A. Khodakovsky, N. Litke, and P. Schröder. Globally smooth parameterizations with low distortion. *ACM Transactions on Graphics*, 22(3):350–357, July 2003.
- [31] S. Kim and S. J. Karrila. *Microhydrodynamics: Principles and Selected Applications*. Butterworth-Heinemann, 1991.
- [32] R. Kress. *Linear Integral Equations*. Applied Mathematical Sciences. Springer, 1999.
- [33] C. Lage and C. Schwab. Wavelet galerkin algorithms for boundary integral equations. *SIAM J. Sci. Comput.*, 20(6):2195–2222 (electronic), 1999. 1095-7197.
- [34] A. Mayo. The fast solution of Poisson’s and the biharmonic equations on irregular regions. *SIAM Journal on Numerical Analysis*, 21(2):285–299, 1984.
- [35] S. G. Mikhlin. *Integral equations and their applications to certain problems in mechanics, mathematical physics and technology*. Second revised edition. Translated from the Russian by A. H. Armstrong. A Pergamon Press Book. The Macmillan Co., New York, 1964.
- [36] K. Nabors, J. Phillips, F. T. Korsmeyer, and J. White. Multipole and precorrected-FFT accelerated iterative methods for solving surface integral formulations of three-dimensional Laplace problems. In *Domain-based parallelism and problem decomposition methods in computational science and engineering*, pages 193–215. SIAM, Philadelphia, PA, 1995.
- [37] N. Nishimura. Fast multipole accelerated boundary integral methods. *Applied Mechanics Review*, 55(4):299–234, 2002.
- [38] H. Power and L. Wrobel. *Boundary Integral Methods in Fluid Mechanics*. Computational Mechanics Publications, 1995.

- [39] C. Pozrikidis. *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge University Press, 1992.
- [40] C. Schwab and W. L. Wendland. Numerical integration of singular and hypersingular integrals in boundary element methods. *Numerical integration (Bergen, 1991)*, 357:203–218, 1992. NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci. Kluwer Acad. Publ. Dordrecht.
- [41] J. Tausch. Sparse BEM for potential theory and stokes flow using variable order wavelets. *Comput. Mech.*, 32(4-6):312–318, 2003. 0178-7675.
- [42] J. Tausch. A variable order wavelet method for the sparse representation of layer potentials in the non-standard form. *J. Numer. Math.*, 12(3):233–254, 2004.
- [43] J. Tausch and J. White. Multiscale bases for the sparse representation of boundary integral operators on complex geometry. *SIAM J. Sci. Comput.*, 24(5):1610–1629 (electronic), 2003. 1095-7197.
- [44] S. Vijayakumar and D. E. Cormack. A new concept in near-singular integral evaluation: the continuation approach. *SIAM J. Appl. Math.*, 49(5):1285–1295, 1989. 0036-1399.
- [45] W. Wintz, H.-G. Döbereiner, and U. Seifert. Starfish vesicles. *Europhysics Letters*, 33:403–408, Feb. 1996.
- [46] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J. Comput. Phys.*, 196(2):591–626, 2004.
- [47] L. Ying, G. Biros, D. Zorin, and H. Langston. A new parallel kernel-independent fast multipole method. In *SC '03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, page 14, Washington, DC, USA, 2003. IEEE Computer Society.
- [48] L. Ying and D. Zorin. A simple manifold-based construction of surfaces of arbitrary smoothness. *ACM Trans. Graph.*, 23(3):271–275, 2004.