Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 219 (2006) 247–275
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains

Lexing Ying [a,*], George Biros [b,c], Denis Zorin [d]

[a] Applied and Computational Mathematics, California Institute of Technology, CA 91125, United States
[b] Department of Mechanical Engineering, University of Pennsylvania, PA 19104, United States
[c] Department of Computer and Information Science, University of Pennsylvania, PA 19104, United States
[d] Courant Institute of Mathematical Sciences, New York University, NY 10012, United States

Received 29 September 2005; received in revised form 6 March 2006; accepted 20 March 2006
Available online 11 May 2006

**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 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. Results from potential theory allow us to represent boundary value problems in integral equation form. 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

* Corresponding author. Tel.: +1 626 395 5760.
*E-mail addresses:* lexing@acm.caltech.edu (L. Ying), biros@seas.upenn.edu (G. Biros), dzorin@cs.nyu.edu (D. Zorin).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.03.021

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 difficulties, especially if the goal is to obtain an algorithm that is asymptotically optimal, accurate, and fast enough to be useful in practical settings. Indeed, in order to get such an algorithm, for domains with smooth boundaries,[1] one has to address five main problems:

- *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 that 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 it is more complicated in the case of three dimensions.
- *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 non-uniform distributed forces, one has to devise efficient schemes for the computation of volume integrals, especially in the case where the support of the function coincides with a volumed domain that has a complex boundary.

In this paper, we present a method that addresses each in turn, save the last.

For two-dimensional boundary value problems in smooth domains, a number of highly efficient boundary integral solvers has been developed [5,21,23,33]. Most implementations are based on indirect formulations that 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 ($C^\infty$ or $C^k$-continuous for sufficiently large $k$, but not necessarily analytic) boundaries, which achieves high-order convergence with linear complexity with respect to 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 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) [49], 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 on 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 [51]. 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 suffi-

[1] Domains with edges and corners present additional challenges that we do not discuss in this particle.

ciently separated from the boundary. Finally, we show how our boundary equation solver can be combined with an FFT-based fast spectral solver on regular grids to obtain solutions of inhomogeneous boundary value problems on domains with 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 that 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 $C^\infty$, 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 $C^\infty$, then $\delta$ can be chosen to be arbitrarily close to zero. To simplify the presentation, we describe our solver for $\delta = 1/2$ in Section 3.

We observe that 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 [5,33,28] 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. One 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 (e.g. 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 spherical and toroidal 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 that is discontinuous but preserves the vanishing moments property) is explored in [1]. In [15,14,34], wavelet bases are constructed on the boundary surfaces directly. As 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 [44–46]. 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 was developed, e.g. [2,43,16,3]; [29] surveys many of the early algorithms. These algorithms focus on the scenario typical for Galerkin methods where 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-based formulations that typically do not involve the definition of a basis.

A number of techniques were used to accelerate inversion of the dense linear system resulting from discretization. These techniques include fast multipole method (FMM) [22], panel clustering method [24], FFT-based approaches [11,38] and, as already mentioned, wavelet-based techniques. FMM runs in linear complexity for any fixed accuracy. A comprehensive survey of algorithms using FMM can be found in [39]. As an important extension of the standard tree code, Panel clustering method has complexity $O(N\log^{d+2} N)$ with $d$ and $N$ being the dimension and the size of the problem, respectively. 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 [47,30,7]. Vijayakumar and Cormack [47] used homogeneity of the kernel to convert the problem to an ODE. Johnson [30] applied a change of coordinates to reduce or move the singularity in the parametric domain. Beale and Lai [7] considered 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 [9,35], 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 their accuracy is limited by the accuracy of the regular grid solver.

Techniques based on integral equation formulations have been widely used in engineering literature to study phenomena related to flows with low Reynolds numbers. For example, Pozrikidis and collaborators [42] describe a number of applications related to flows of suspensions of liquid capsules. Zinchenko and Davis [52] developed an efficient algorithm for computing hydrodynamical interaction of deformable droplets with reasonable accuracy. Muldowney and Higdon [37] introduced a spectral element approach for 3D Stokes flow, which was applied to compute the resistance functions for particles in cylindrical domains [27]. However, most numerical examples there were limited to simple geometries.

### 1.2. Boundary value problems

We consider three elliptic equations: the Laplace equation, the Stokes equation, and the Navier equation. The Laplace equation on an open set $\Omega$ with boundary $\Gamma$ 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:

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

where $\boldsymbol{u}$ is the velocity, $p$ is the pressure, $\mu$ is a positive constant and the equation $\operatorname{div}\boldsymbol{u} = 0$ is often called the *incompressibility condition*.

Finally, the Navier equation is:

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

where $\boldsymbol{u}$ is the displacement field, $\mu$ 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 $\Gamma$ is the union of *overlapping*

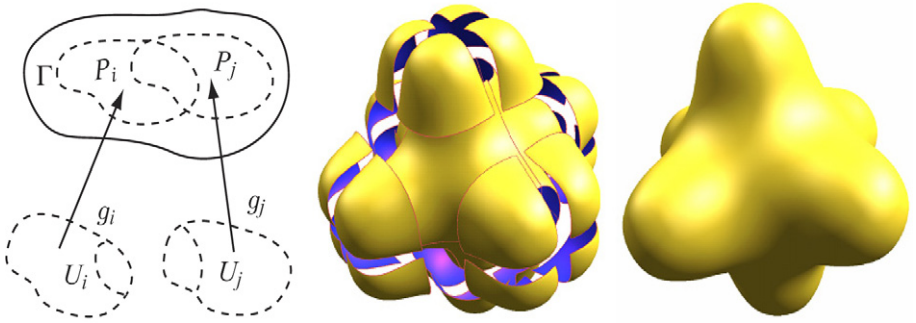



Fig. 1. Left: Parameterization of the boundary. The middle and right frames illustrate an example surface. Middle: Multiple patches shown separately. Right: The surface is the union of all the patches.

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 $g_k : U_k \to \mathbb{R}^3$ and $g_k(U_k) = P_k$ (see Fig. 1). We assume $g_k$ and the manifold structure to be $C^\infty$; this assumption can be relaxed to $C^k$ for a sufficiently large $k$ (how large $k$ should be depends the desired convergence rate). It is important to note that the surface in this case is *not* defined as a collection of separate and disjoint parameterizations.

Most of the analytic surfaces, such as spheres, torus, and ellipsoids, have simple manifold representations. In [51], we describe a framework to construct general smooth surfaces with explicit manifold representations from arbitrary coarse meshes, along with implementation details. Reparametrization techniques (e.g. [31]) can be used to construct a manifold-type representation from an arbitrary fine triangle mesh.

### 1.4. Organization of the paper

In Section 2, we start with a review of the boundary integral formulation of the related elliptic PDE problems and describe briefly 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 discretization and quadrature rule for the singular integrals in the boundary integral formulation. Section 4 presents a new algorithm for integrating nearly singular integrals that appear in the evaluation of the solution at points that 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.

For the remainder of the paper, we describe our method in the case of the Stokes equations, and with respect to the remaining two equations, we discuss the necessary modifications to our 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 [36,32,40]. The Stokes equations:

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

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

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

where $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}$, $\boldsymbol{n}(\boldsymbol{y})$ is the exterior normal direction at $\boldsymbol{y}$, $|\boldsymbol{r}|$ is the Euclidean norm of $\boldsymbol{r}$; $D$ is weakly-singular and called the double layer kernel for the velocity field, and the function $\boldsymbol{\phi}$, defined on $\Gamma$, is called the double

layer density. We often write (2) as $(D\boldsymbol{\phi})(\boldsymbol{x})$. If we momentarily ignore the incompressibility condition (i.e., $\operatorname{div}\boldsymbol{u} = 0$) and assume the domain $\Omega$ to be simply-connected and bounded, we can obtain the solution of (1) by solving for the density $\boldsymbol{\phi}$ from

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

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

$$\boldsymbol{u}(\boldsymbol{x}) = \int_\Gamma D(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\phi}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}) \quad \text{and} \quad p(\boldsymbol{x}) = \int_\Gamma K(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\phi}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}), \tag{4}$$

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

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{1}{2}[[\boldsymbol{u}]](\boldsymbol{x}) + \int_\Gamma D(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\phi}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}) \quad \text{and} \quad p(\boldsymbol{x}) = \frac{1}{2}[[p]](\boldsymbol{x}) + \int_\Gamma K(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\phi}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}), \tag{5}$$

where $[[\boldsymbol{u}]](\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})$ is the difference between the limit values of the velocity fields inside and outside the domain at $\boldsymbol{x} \in \Gamma$, and $[[p]](\boldsymbol{x})$ is the difference of the pressure field (see Appendix B for the exact formula). The integral for pressure in (5) is understood in the *Hadamard* sense (see p. 264 of [23]).

To account for incompressibility of the fluid and domain geometry, we need to modify these basic steps. We consider three different types of domains, shown in Fig. 2.

### 2.1. Single-surface bounded domains

In this case, we need to include the incompressibility condition. Following [32,40], we obtain $\boldsymbol{\phi}$ by solving a modified version of (3)

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

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

### 2.2. Unbounded domains

Suppose $\Gamma = \bigcup_{m=1}^{M}\Gamma_m$ where $\Gamma_m$ 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 $\Omega$. Following [21,32,40], the solution is to introduce additional *Stokeslet* and *Rotlet* terms for each $\Gamma_m$ for $1 \leqslant m \leqslant M$. The Stokeslet is the Green's function of the Stokes equations:

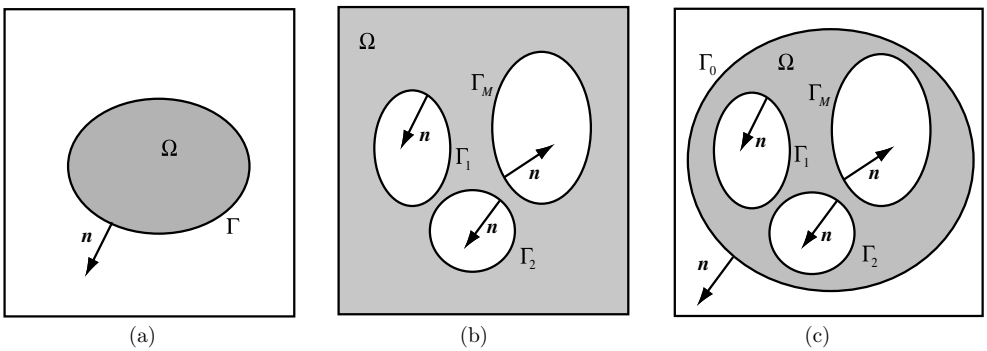



Fig. 2. Domain types: (a) single-surface bounded domain; (b) unbounded domain; (c) multiple-surface bounded domain.

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

The Rotlet is defined by

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

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

$$\begin{cases} \frac{1}{2}\boldsymbol{\phi}(\boldsymbol{x}) + (D\boldsymbol{\phi})(\boldsymbol{x}) + \sum_{m=1}^{M}(S(\boldsymbol{x}-\boldsymbol{z}_m)\alpha_m + R(\boldsymbol{x}-\boldsymbol{z}_m)\beta_m) = \boldsymbol{b}(\boldsymbol{x}) \quad (\boldsymbol{x}\in\Gamma), \\ \int_{\Gamma_m}\boldsymbol{\phi}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}) = 0 \quad (1\leqslant m\leqslant M), \\ \int_{\Gamma_m}(\boldsymbol{y}-\boldsymbol{z}_m)\times\boldsymbol{\phi}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}) = 0 \quad (1\leqslant m\leqslant M). \end{cases} \tag{7}$$

Then, the velocity and pressure are computed by:

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

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

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

for any $\boldsymbol{g}$. Here the pressure does not depend on $\beta_m$ since any Rotlet generates zero pressure.

### 2.3. Multiple-surface bounded domains

Suppose the boundary $\Gamma$ consists of $M+1$ connected components $\Gamma_m$, $m = 0,\ldots,M$, and $\Gamma_0$ encloses all other components $\Gamma_m$ for $m = 1,\ldots,M$. We first solve for $\boldsymbol{\phi}$, $\alpha_m$ and $\beta_m$ using:

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

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

**Remark 2.1.** The integral equations for the Navier equations are very similar to those of 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:

$$\begin{cases} \frac{1}{2}\boldsymbol{\phi}(\boldsymbol{x}) + (D\boldsymbol{\phi})(\boldsymbol{x}) + \sum_{m=1}^{M}S(\boldsymbol{x}-\boldsymbol{z}_m)\alpha_m = \boldsymbol{b}(\boldsymbol{x}) \quad (\boldsymbol{x}\in\Gamma), \\ \int_{\Gamma_m}\boldsymbol{\phi}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}) = 0 \quad (1\leqslant m\leqslant M), \end{cases}$$

where $S$ is the Green's function of the Laplace equation and $\phi$ and $\alpha_m$ 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

$$(D\boldsymbol{\phi})(\boldsymbol{x}) = \int_{\Gamma} D(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\phi}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}) \quad (\boldsymbol{x}\in\Gamma). \tag{10}$$

This integral needs to be computed both 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

As noted in Section 1.3, the domain boundary $\Gamma$ is the union of a set of *overlapping* patches $P_k$, $k = 1,\ldots,K$, smoothly parameterized over a chart $U_k$ by smooth functions $g_k$ (hereinafter, we use "smooth" to describe $C^\infty$ or $C^k$ functions for $k$ sufficiently large not to affect the convergence estimates). Additionally, we utilize a partition of unity $\{w_k\colon \Gamma \to R\}$ satisfying the following conditions:

- Each $w_k$ is smooth and non-negative in $P_k$, and vanishes on the boundary of $P_k$.
- $\sum_{k=1}^{K} w_k(\boldsymbol{x}) = 1$ for all $\boldsymbol{x}\in\Gamma$.

As we will see in this section, the overlapping patches and the partition of unity allows us to use the trapezoidal rule to compute the smooth part of (10) with high-order accuracy. For surfaces constructed by the method in [51], the partition of unity used for surface construction naturally satisfies the conditions above (see [51] for details). For general surfaces represented as overlapping patches, it is often not difficult to construct $\{f_k\}$ that satisfy the first condition. Then, we can simply choose $\{w_k = f_k/(\sum_{k=1}^{K} f_k)\}$ as the partition of unity.

### 3.1. Discretization and quadrature rules

As we integrate singular kernels, we need to use high-accuracy quadrature rules for singular functions. Though 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. In the alternative, one can choose regular quadrature point placements, and take advantage of domain transformations to adapt quadrature points to the singularity. Our approach is based on the development in [11] for the Helmholtz equation.

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

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

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

$$\boldsymbol{\psi}_k(c_k) = w_k(g_k(c_k))\boldsymbol{\phi}(g_k(c_k))J_k(c_k).$$

Since $w_k$ vanishes at the boundary of $P_k$, both $w_k(g_k(\cdot))$ and $\boldsymbol{\psi}_k(\cdot)$ vanish at the boundary of $U_k$. Using this notation, the integral becomes

$$(D\boldsymbol{\phi})(\boldsymbol{x}) = \sum_{k=1}^{K} \int_{U_k} D(\boldsymbol{x}, g_k(c_k))\boldsymbol{\psi}_k(c_k)\,\mathrm{d}c_k. \tag{11}$$

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 $D\boldsymbol{\phi}(\boldsymbol{x})$.

In subsequent formulas, we use $\boldsymbol{\phi}_{k,i}$ to denote the value of function $\boldsymbol{\phi}$ at point $c_{k,i}$. Similarly, $\boldsymbol{\psi}_{k,i}$ is the value of $\boldsymbol{\psi}_k$ at $c_{k,i}$ and $\boldsymbol{x}_{k,i}$ 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

$$\int_U D(\boldsymbol{x}, g(c))\boldsymbol{\psi}(c)\,\mathrm{d}c. \tag{12}$$

*3.1.1. Non-singular integration*

If $\boldsymbol{x} \notin g(U)$, then $D(\boldsymbol{x}, g(c))$ is non-singular for any point $c \in U$, and the integrand $D(\boldsymbol{x}, g(c))\boldsymbol{\psi}(c)$ 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 periodic domains. For $C^M$ functions, the trapezoidal rule converges as $\mathrm{O}(h^M)$.

*3.1.2. Singular integration*

If $\boldsymbol{x} = g(c')$ for some $c' \in U$, we introduce a $C^\infty$ function $\eta_{c'}$ defined by

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

where $\chi\colon [0,\infty) \to [0,1]$ is a non-increasing $C^\infty$ function satisfying $\chi(r) = 1$ in a neighborhood of zero and $\chi(r) = 0$ for $r \geqslant 1$. In practice, we choose $\chi$ such that $\chi(r) = 1$ for $r \leqslant 1/4$. 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 $\boldsymbol{x}$. Using the function $\eta_{c'}$, we split the integral (12) into two parts:

$$\int_U D(g(c'), g(c))\boldsymbol{\psi}(c)\,\mathrm{d}c = \int_U D(g(c'), g(c))(1 - \eta_{c'}(c))\boldsymbol{\psi}(c)\,\mathrm{d}c \tag{13}$$

$$+ \int_U D(g(c'), g(c))\eta_{c'}(c)\boldsymbol{\psi}(c)\,\mathrm{d}c. \tag{14}$$

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

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

In this formula, we allow $\rho$ to be negative by restricting the integration domain of $\theta$ to be $[0,\pi)$. It can be shown (see Appendix A) that for a fixed $\theta$, $\rho D(g(c'), g(c(\rho,\theta)))$ is a smooth function of $\rho$. 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 $\theta$, the double integral (15) can be integrated using the trapezoidal rule again. The coordinates of the quadrature points in the polar coordinate system $(\rho,\theta)$ are

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

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

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 $\mathrm{O}(h^M)$ for any integer $M$.

*3.1.3. Density interpolation*

The quadrature points for (14) are on a Cartesian grid in polar coordinates that do not coincide with the points at which the unknown function is discretized. Therefore, an interpolation procedure is necessary to

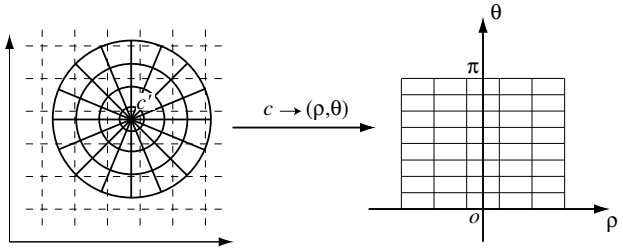



Fig. 3. Integration points in Cartesian (left) and polar (right) coordinates.

obtain the values of $\psi$ 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.

In the preprocessing step, we compute an additional set of values of $\psi$ on the chart $U$. Since $\psi$ 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 $\psi$ from the values $\psi_i$ at $c_i$. We then use these Fourier coefficients to approximate the values of $\psi$ on a new grid, which is $m$ times finer than the original grid, by means of an inverse FFT. Finally, a bicubic B-spline interpolant is constructed from the values of $\psi$ on this refined grid with periodic end conditions.

Given a point $c \in U$, we need to compute the values of $\psi$ at a polar-coordinate grid centered at $c$. This is done by simply evaluating the B-spline interpolant constructed in the preprocessing step.

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

Our quadrature rules have two important novel aspects: (1) the choice of the size of the support of the floating partition of unity function $\eta$ 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 Fig. 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 and without compromising accuracy, by using the kernel-independent fast multipole method.

For each point $\boldsymbol{x}$ on the boundary $\Gamma$, we need to evaluate the integral

$$(D\boldsymbol{\phi})(\boldsymbol{x}) = \sum_{k=1}^{K} \int_{U_k} D(\boldsymbol{x}, g_k(c_k))\boldsymbol{\psi}_k(c_k)\,\mathrm{d}c_k,$$

which is decomposed into a sum of three parts:

$$\sum_{k:\boldsymbol{x}\notin g_k(U_k)} \int_{U_k} D(\boldsymbol{x}, g_k(c_k))\boldsymbol{\psi}_k(c_k)\,\mathrm{d}c_k, \tag{16}$$

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

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

where $c'_k = g_k^{-1}(\boldsymbol{x})$ 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

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

where $\boldsymbol{\psi}_{k,i} = \psi_k(c_{k,i})$. In the first summation extending over the whole boundary surface, the terms $\boldsymbol{\psi}_{k,i}h^2$ are independent of the evaluation point $\boldsymbol{x}$. Therefore, we can use the kernel-independent fast multipole method developed in [49] to evaluate the value for all quadrature points $\boldsymbol{x}$ efficiently without compromising 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 $\boldsymbol{d}_{l,j} = (D\boldsymbol{\phi})(\boldsymbol{x}_{l,j})$, 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**
  $\psi_{k,i} \leftarrow w_{k,i}\phi_{k,i}J_k(c_{k,i})$
**end for**
{Step 1: Add terms of (19) with evaluation point independent weights.}
Set all $d_{l,j}$ to approximation of $\sum_{k,i} D(\boldsymbol{x}_{l,j}, \boldsymbol{x}_{k,i})\boldsymbol{\psi}_{k,i}h^2$ 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 $\boldsymbol{x}_{l,j} = g_k(c'_k)$ for some $c'_k \in U_k$ **do**
    $\boldsymbol{d}_{l,j} \leftarrow \boldsymbol{d}_{l,j} - \sum_{i:\psi_{k,i}>0} D(\boldsymbol{x}_{l,j}, g_k(c_{k,i}))\eta_{c'_k}(c_{k,i})\boldsymbol{\psi}_{k,i}h^2$
  **end for**
**end for**
{Step 3: Preprocess the grids $\psi_{k,i}$ for high-order interpolation.}
{Step 4: Add (18).}
**for all** $(l,j)$ **do**
  **for all** $U_k$ such that $\boldsymbol{x}_{l,j} = g_k(c'_k)$ for some $c'_k \in U_k$ **do**
    Add to $d_{l,j}$ the discretization of $\int_{U_k} D(\boldsymbol{x}_{l,j}, g_k(c_k))\eta_{c'_k}(c_k)\boldsymbol{\psi}_k(c_k)\,\mathrm{d}c_k$ using polar coordinates integration.
  **end for**
**end for**

### 3.2.1. Complexity analysis

Let $N$ be the total number of quadrature points $\boldsymbol{x}_{k,i}$. 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 $\Gamma$, 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 $\epsilon_{\text{FMM}}$.
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 $\boldsymbol{x}_{k,i}$ 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 the costs of all four stages, 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 shows that the error is $O(h^{\frac{M-1}{2}})$ if the double layer density $\boldsymbol{\phi}$ is $C^M$. If, instead of

using the radius proportional to $\sqrt{h}$, we use the radius $h^{1-\gamma}$ for $0 < \gamma \leqslant 1$, we obtain an algorithm with complexity $N^{1+\gamma}$. For example, by using a partition of unity that 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 that 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 $\boldsymbol{\phi}_{k,i}$ to the vector of $\boldsymbol{d}_{l,j}$. Hereinafter $Y^D$ denotes this linear operator. To summarize,

- Evaluation of $Y^D$ has complexity $O(N^{3/2})$.
- The approximation error $(Y^D\boldsymbol{\phi} - D\boldsymbol{\phi})(\boldsymbol{x})$, when $\boldsymbol{\phi}$ is a $C^M$ function, is bounded by $\max(C_1 h^{\frac{M-1}{2}}, C_2\epsilon_{\mathrm{FMM}}, C_2\epsilon_{\mathrm{interp}})$ for $\boldsymbol{x}$ on $\Gamma$, where the constant $C_1$ depends on the $M$th order derivatives of $\boldsymbol{\phi}$ and $C_2$ is a bound on the $L^\infty$ norm of $D\boldsymbol{\phi}$. As previously mentioned, $\epsilon_{\mathrm{FMM}}$ is the error bound of the kernel-independent FMM and $\epsilon_{\mathrm{interp}}$ 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 that of 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, without necessitating any change in the algorithm.

### 3.3. Hypersingular pressure evaluation

In this section, we describe the algorithm used to evaluate the pressure value $p$ on the boundary $\Gamma$ from the solution $\boldsymbol{\phi}$ using (5). This is an essential step in our algorithm for the evaluation of $\boldsymbol{u}$ and $p$ in the interior of the domain $\Omega$ (see Section 4).

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

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

where $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}$ and $\boldsymbol{n}(\boldsymbol{y})$ 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 $|\boldsymbol{r}|^{-3}$ (i.e., the integral is hypersingular). Evaluating these integrals requires modification to our algorithm.

In order to derive the formula for $p(\boldsymbol{x})$ for $\boldsymbol{x}$ on the boundary $\Gamma$, we use the following fact (from potential theory for the Stokes operator): if $\boldsymbol{\phi} \equiv c$ is a constant, then the velocity field $\boldsymbol{u}$ in $\Omega$ generated by $\boldsymbol{\phi}$ is again a constant, and correspondingly, the pressure field $p$ is zero [33,40,41]. Assume $\boldsymbol{x}' \in \Omega$ approaches a boundary point $\boldsymbol{x} \in \Gamma$, in which case we can write the formula for the pressure in the following form:

$$p(\boldsymbol{x}') = \int_\Gamma K(\boldsymbol{x}', \boldsymbol{y})\cdot(\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\phi}(\boldsymbol{x}))\,\mathrm{d}s(\boldsymbol{y}).$$

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

$$p(\boldsymbol{x}) = \frac{1}{2}[[p]](\boldsymbol{x}) + \int_\Gamma K(\boldsymbol{x}, \boldsymbol{y})\cdot(\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\phi}(\boldsymbol{x}))\,\mathrm{d}s(\boldsymbol{y}), \tag{20}$$

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

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

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

where $\boldsymbol{\phi}_{\boldsymbol{\alpha}}$ and $\boldsymbol{\phi}_{\boldsymbol{\beta}}$ denote the directional derivatives of $\boldsymbol{\phi}$ in the directions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively.

*3.3.1. Jump evaluation*

To evaluate the jump $[[p]](\boldsymbol{x})$, we need to compute $\boldsymbol{\phi}_{\boldsymbol{\alpha}}$ for a direction of $\boldsymbol{\alpha}$ in the tangent plane at $\boldsymbol{x}$ from the values of $\boldsymbol{\phi}$ at the quadrature points $\boldsymbol{x}_{k,i}$. The most straightforward approach would be to use the 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 $\boldsymbol{\phi}$ is not compactly supported on each parametric domain, directly interpolating the directional derivatives of $\boldsymbol{\phi}$ often results low order accuracy at points close to the boundary of the parametric domain. To achieve high-order approximation to the directional derivatives $\boldsymbol{\phi}_{\boldsymbol{\alpha}}$ and $\boldsymbol{\phi}_{\boldsymbol{\beta}}$, we use the partition of unity again. Using $\boldsymbol{\phi}_{\boldsymbol{\alpha}}$ as an example, we write

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

Here, $w_k\boldsymbol{\phi}$ 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 $\boldsymbol{\psi}$. Again, an FFT-based preprocessing step is used to build a B-spline interpolant on an eight-fold refined grid for each directional derivative. At the evaluation stage, we simply evaluate the interpolant to approximate the value of $(w_k\boldsymbol{\phi})_{\boldsymbol{\alpha}}$.

*3.3.2. 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 $\boldsymbol{\phi} - \boldsymbol{\phi}^{\boldsymbol{x}}$, where $\boldsymbol{\phi}^{\boldsymbol{x}}(\boldsymbol{y})$ is a constant density with value $\boldsymbol{\phi}(\boldsymbol{x})$ for any $\boldsymbol{y} \in \Gamma$. However, $\boldsymbol{\phi} - \boldsymbol{\phi}^{\boldsymbol{x}}$ depends on the target point $\boldsymbol{x}$, and the result $Y^K(\boldsymbol{\phi} - \boldsymbol{\phi}^{\boldsymbol{x}})$ only gives the valid pressure value at the point $\boldsymbol{x}$. Clearly applying the operator $Y^K$ to $\boldsymbol{\phi} - \boldsymbol{\phi}^{\boldsymbol{x}}$ for each $\boldsymbol{x}$ is prohibitively expensive. The algorithm we propose uses the linearity of $Y^K$ to evaluate $(K\boldsymbol{\phi})(\boldsymbol{x})$ at all points $\boldsymbol{x}$ simultaneously and much more efficiently.

Observe that

$$(Y^K(\boldsymbol{\phi} - \boldsymbol{\phi}^{\boldsymbol{x}}))(\boldsymbol{x}) = (Y^K\boldsymbol{\phi})(\boldsymbol{x}) - (Y^K\boldsymbol{\phi}^{\boldsymbol{x}})(\boldsymbol{x}) = (Y^K\boldsymbol{\phi})(\boldsymbol{x}) - ((Y^K\boldsymbol{e}^1)(\boldsymbol{x}), (Y^K\boldsymbol{e}^2)(\boldsymbol{x}), (Y^K\boldsymbol{e}^3)(\boldsymbol{x})) \cdot \boldsymbol{\phi}(\boldsymbol{x}).$$

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

**Algorithm 2** (*Singular integration for the pressure*).

Evaluate $g^d = Y^K\boldsymbol{e}^d$ for $d = 1, 2, 3$.
Evaluate $p = Y^K\phi$.
**for** each evaluation point $x$ **do**
  $p(x) \leftarrow p(x) - (g^1(x), g^2(x), g^3(x)) \cdot \phi(x)$.
**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 $\boldsymbol{\phi}$. As the discretization is refined, however, convergence of the approximation to the correct values depends on cancellation: both values $Y^K\boldsymbol{\phi}$ and $Y^K\boldsymbol{\phi}^{\boldsymbol{x}}$ at point $\boldsymbol{x}$ may increase, however, their difference approaches the correct limit value. This indicates that the algorithm potentially may suffer from floating point errors due to the cancellation of large quantities. However, by using double precision arithmetics, we have not observed the degradation of the accuracy in our numerical experiments.

## 4. Nearly singular integration

In this section, we present the algorithm to evaluate the velocity $\boldsymbol{u}$ and the pressure $p$ at an arbitrary point $\boldsymbol{x}$ in the domain.

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

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

The integrand is not singular, since $D(\boldsymbol{x},\boldsymbol{y})$ is not singular when $\boldsymbol{x} \notin \Gamma$. If the distance from $\boldsymbol{x}$ to $\Gamma$ 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 $U_k$, $k = 1,\ldots,K$ with evenly spaced quadrature points $c_{k,i}$ has optimal accuracy. However, as $\boldsymbol{x}$ approaches the boundary $\Gamma$, $D(\boldsymbol{x},\boldsymbol{y})$ becomes nearly singular and oscillatory, and the derivatives of $D(\boldsymbol{x},\boldsymbol{y})$ with respect to $\boldsymbol{y}$ cannot be bounded uniformly.

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

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

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

If $\boldsymbol{x}$ is in the *well-separated* region $\Omega_0$, we use the trapezoidal rule on the Nyström points $\boldsymbol{x}_{k,i}$ with weights $\boldsymbol{\psi}_{k,i}$ to evaluate $(D\boldsymbol{\phi})(\boldsymbol{x})$, where $\boldsymbol{\psi}_{k,i}$ are defined as in Section 3.2.

For the points $\boldsymbol{x}$ in the *intermediate* region $\Omega_1$, for each chart $U_k$, we resample the function $\boldsymbol{\psi}_k$ on a refined Cartesian grid with spacing $h^{3/2}$ using the values $\boldsymbol{\psi}_{k,i}$ at the original grid points $\boldsymbol{x}_{k,i}$ with grid step $h$. Since both grids are Cartesian and $\boldsymbol{\psi}_k$ is periodic function on $U_k$, the resampling can be done first by computing FFT of $\boldsymbol{\psi}_k$, then padding the higher frequency with zeros, and finally using inverse FFT to compute the values on the refined grid. With $\boldsymbol{\psi}_k$'s values on the refined grid available, we use the trapezoidal rule on the new grid to evaluate $D\boldsymbol{\phi}(\boldsymbol{x})$. 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 $\Omega_2$, we interpolate between the values at points on the surface $\Gamma$ and points in $\Omega_1$ or $\Omega_0$. For each point $\boldsymbol{x}$ in $\Omega_2$, we first find a point $\boldsymbol{x}_0 \in \Gamma$ such that

$$\frac{\boldsymbol{x}_0 - \boldsymbol{x}}{|\boldsymbol{x}_0 - \boldsymbol{x}|} \cdot \boldsymbol{n}(\boldsymbol{x}_0) \geqslant \alpha, \tag{21}$$

where the value of $\alpha$ is less than but close to 1, which means that $\boldsymbol{x} - \boldsymbol{x}_0$ is almost orthogonal to the tangent plane at $\boldsymbol{x}_0$. The algorithm is not very sensitive to the value of $\alpha$; we use $\alpha = 0.95$ in our implementation. We find $\boldsymbol{x}_0$ 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 $\boldsymbol{x}_0$ in 3 or 4 iterations.

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

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

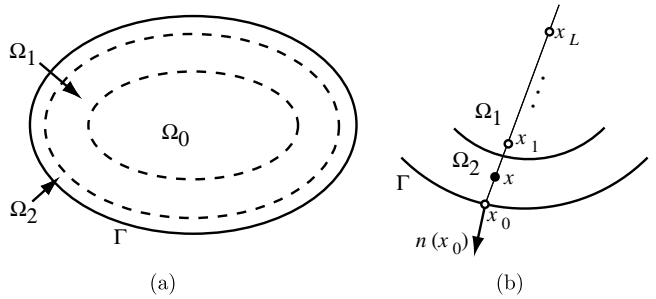



Fig. 4. Evaluation of nearly singular integrals. (a) Different regions based on their distance to the boundary. (b) Evaluation procedure for $\boldsymbol{x} \in \Omega_2$.

where $\beta$ is a constant such that $\alpha\beta$ is greater than but close to 1. Since $\alpha$ is close to 1, we can choose $\beta$ to be close to 1, as well. We then use the singular quadrature algorithm described in Section 3 to evaluate $\frac{1}{2}\boldsymbol{\phi}(\boldsymbol{x}_0) + D\boldsymbol{\phi}(\boldsymbol{x}_0)$, which is the limit of $D\boldsymbol{\phi}$ at $\boldsymbol{x}_0$. The points $\{\boldsymbol{x}_l,\ l = 1,\ldots,L\}$ are now in $\Omega_1$ or $\Omega_0$ since $\alpha\beta \geqslant 1$, and we evaluate $\{(D\boldsymbol{\phi})(\boldsymbol{x}_l),\ l = 1,\ldots,L\}$ using the procedure described above for regions $\Omega_1$ and $\Omega_0$. Finally, we use these values at $\{\boldsymbol{x}_l,\ l = 0,\ldots,L\}$ to perform a 1D Lagrange interpolation of order $L$ to obtain the value of $D\boldsymbol{\phi}$ at $\boldsymbol{x}$ (see Fig. 4(b)). The order of interpolation is chosen to achieve the desired convergence rate, as discussed in Appendix C.

### 4.1. Summary and complexity analysis

To evaluate the velocity on $O(N^{3/2})$ points that uniformly sample $\Omega$ with approximately the same discretization used for the boundary, we first identify for each point to which region ($\Omega_0$, $\Omega_1$ or $\Omega_2$) it belongs and then use the corresponding algorithm described in the previous section.

- There are $O(N^{3/2})$ points in the well-separated region $\Omega_0$. We evaluate the velocity using the trapezoidal rule on the quadrature points $\boldsymbol{x}_{k,i}$ with FMM acceleration. This step has complexity $O(N^{3/2})$.
- There are $O(N^{5/4})$ points in intermediate region $\Omega_1$. We first interpolate $\boldsymbol{\phi}$ 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 $\Omega_2$. For each point $\boldsymbol{x}$ in $\Omega_2$, we first find the correspondent $\boldsymbol{x}_0$ and $\boldsymbol{x}_l$ for $l = 1,\ldots,L$. The evaluation of $\frac{1}{2}\boldsymbol{\phi}(\boldsymbol{x}_0) + (D\boldsymbol{\phi})(\boldsymbol{x}_0)$ for all points $\{\boldsymbol{x}_0: \boldsymbol{x} \in \Omega_2\}$ can be achieved by using the algorithm in Section 3. We then evaluate $\{(D\boldsymbol{\phi})(\boldsymbol{x}_l): \boldsymbol{x} \in \Omega_2,\ l = 1,\ldots,L\}$ using the refined grid with FMM acceleration. The complexity of the evaluation for $\{\boldsymbol{x} \in \Omega_2\}$ is $O(N^{3/2})$ again.

Summing up the costs of all three regions, we conclude that the complexity of the complete algorithm is $O(N^{3/2})$.

In Appendix C, we give a proof for the convergence rate 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 $\boldsymbol{x}$ in $\Omega$ is similar to the presented algorithm for $\boldsymbol{u}$. The only difference is that since $p$ contains a higher-order singularity. Therefore, the error bound for the velocity of points in regions $\Omega_1$ and $\Omega_2$ decreases more slowly with $h$, and this is observed in practice. To improve the accuracy, we use singularity subtraction: for a point $\boldsymbol{x}$ in $\Omega_1$ or $\Omega_0$ that is close to $\Gamma$, we find its nearest point $\boldsymbol{x}_0$ on the boundary. Then instead of evaluating the pressure using $\int_\Gamma K(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\phi}(\boldsymbol{y})\,ds(\boldsymbol{y})$, we use

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

where $\boldsymbol{\phi}^{\boldsymbol{x}_0}(\boldsymbol{y}) = \boldsymbol{\phi}(\boldsymbol{x}_0)$ for any $\boldsymbol{y}$. Similarly in the case of singular quadrature, the order of singularity is reduced. Efficient implementation for the second integral tracks 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:

$$\begin{cases} -\mu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{f} & \text{in } \Omega, \\ \operatorname{div}\boldsymbol{u} = 0 & \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{b} & \text{on } \Gamma, \end{cases} \tag{22}$$

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

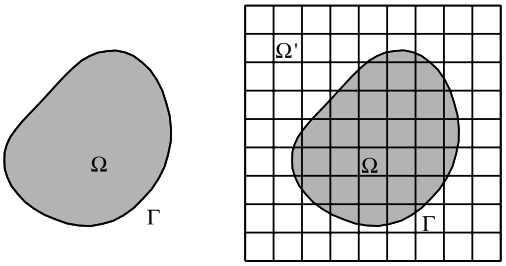



Fig. 5. Domains $\Omega$ and $\Omega'$.

We assume that $\boldsymbol{f}$ is defined on a rectangular domain $\Omega'$, which contains $\Omega$. Even if $\boldsymbol{f}$ is defined only on $\Omega$, it is typically possible to extend $\boldsymbol{f}$ onto $\Omega'$. The domain $\Omega'$ 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 $\Omega$ (see Fig. 5). The method can be extended to adaptively refined Cartesian grids.

The embedded boundary integral method splits the solution of the problem into several steps. We start by choosing a function $\eta$ defined on $\Omega'$, which satisfies two conditions:

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

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

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

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

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

The second problem is:

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

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

The last problem is a homogeneous problem over $\Omega$:

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

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

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

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

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

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

## 6. Numerical results

### 6.1. 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 2 GHz CPU. The boundary of the domains of the test examples was represented using either analytic parameterization or the method of Ying and Zorin [51]. $L$, the number of extra points introduced for each nearby evaluation in the nearest region $\Omega_2$ (see Section 4), is set to be 4 in all examples.

### 6.2. 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. We set the accuracy of the kernel-independent FMM to be smaller compared to the quadrature error from integrating the singularity. Unless otherwise noted, the FMM accuracy is equal to $10^{-8}$.

Table 1 shows the results for the Laplace equation on a bounded domain. The boundary surface is shown in Fig. 6(c). The special solution we chose is $u(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in \Omega$. 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

Table 1
Singular and nearly-singular integration of the Laplace kernel

| $h$ | $N$ | $T_s$ | $\text{Error}_s$ | $\text{Rate}_s$ | $T_n$ | $\text{Error}_n$ | $\text{Rate}_n$ |
|---|---|---|---|---|---|---|---|
| 0.10000 | 1688 | 1.660e + 00 | 7.551e − 04 | 3.3604 | 5.100e − 01 | 5.259e − 04 | 2.8486 |
| 0.05000 | 5480 | 8.980e + 00 | 7.352e − 05 | 3.7527 | 2.950e + 00 | 7.301e − 05 | 3.5945 |
| 0.02500 | 20,736 | 6.928e + 01 | 5.456e − 06 | 2.8887 | 2.229e + 01 | 6.046e − 06 | 2.9417 |
| 0.01250 | 78,792 | 4.636e + 02 | 7.367e − 07 | 3.1928 | 1.039e + 02 | 7.869e − 07 | 3.1084 |
| 0.00625 | 309,520 | 3.202e + 03 | 8.057e − 08 | | 6.443e + 02 | 9.124e − 08 | |

Columns: $h$ is the spacing of discretization, $N$ is the number of Nyström points, $T_s$ is the time spent on singular evaluation, $\text{Error}_s$ is the error of singular evaluation, $\text{Rate}_s$ is the rate of convergence estimated from the errors of two consecutive singular evaluations, $T_n$ is the time spent on nearly-singular evaluation, $\text{Error}_n$ is the error of nearly-singular evaluation, and finally, $\text{Rate}_n$ the rate of convergence estimated from the errors of two consecutive nearly-singular evaluations. In the last run, the FMM accuracy is set to be $10^{-10}$ in order to make it substantially smaller than the quadrature error from singular integration.

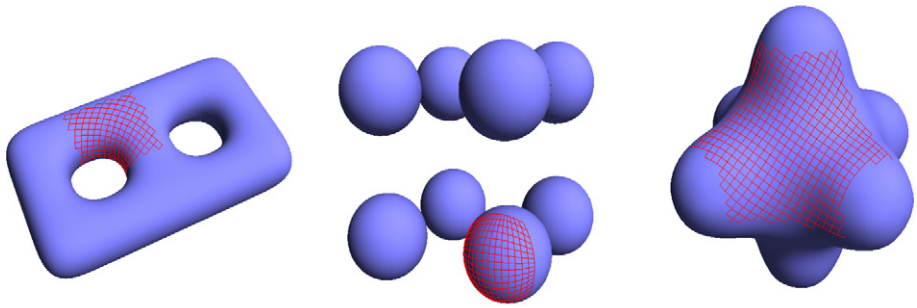



Fig. 6. Boundary surfaces of the test domains. For each plot, the grid shows the Nyström points in a single patch.

double layer density. To test the nearly-singular quadrature algorithm, we choose $N$ points from the interior of the domain, which are very close to the surface.

For the Stokes equations, we use the unit ball as the test domain. The special solution we use is the rigid body fluid rotation. The associated velocity field is $\boldsymbol{u}(\boldsymbol{x}) = \omega \times \boldsymbol{x}$ for $\boldsymbol{x} \in \Omega$ where $\omega$ is the angular velocity vector of the rotation and has unitary norm. In this case, the double layer density $\phi(\boldsymbol{x})$ is equal to $\omega \times \boldsymbol{x}$, 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.

The convergence rate of both examples is mostly between 3 and 4. This rate is primarily determined by the radius of the floating partition of unity, which is set to be $O(\sqrt{h})$ in these examples. Extra numerical tests show that a floating partition of unity with radius $O(h^{1/4})$ does increase the convergence rate substantially, but at the cost of higher computational complexity.

### 6.3. 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 with restart number equal to 40 and halting relative tolerance equal to $10^{-9}$. The FMM accuracy is again set to be $10^{-8}$ in most runs, except a few with high resolution grids, for which the FMM accuracy is equal to $10^{-10}$. In every test, we embed the domain into a dense regular grid with cell size matching the spacing $h$ used to discretize the boundary of the domain. As a result, the distances between many Cartesian grid points 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.

**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 Fig. 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.

Table 2
Singular and nearly-singular integration of the Stokes kernel

| $h$ | $N$ | $T_s$ | $\text{Error}_s$ | $\text{Rate}_s$ | $T_n$ | $\text{Error}_n$ | $\text{Rate}_n$ |
|---|---|---|---|---|---|---|---|
| 0.1000 | 2400 | 2.410e + 00 | 3.489e − 04 | 2.9958 | 4.460e + 00 | 5.072e − 04 | 3.3098 |
| 0.0500 | 9600 | 1.621e + 01 | 4.374e − 05 | 3.9773 | 2.113e + 01 | 5.115e − 05 | 4.0126 |
| 0.0250 | 38,400 | 1.228e + 02 | 2.778e − 06 | 3.5311 | 1.664e + 02 | 3.169e − 06 | 3.7214 |
| 0.0125 | 149,784 | 9.142e + 02 | 2.403e − 07 | | 1.333e + 03 | 2.403e − 07 | |

The headers for the columns are the same as in Table 1.

Table 3
Example 1. Laplace equation on a bounded domain with boundary shown in Fig. 6(a)

| $h$ | $N$ | Iterations | FMM(s) | Local(s) | Total(s) | Error | Rate |
|---|---|---|---|---|---|---|---|
| 0.1000 | 2158 | 11 | 2.698e − 01 | 1.565e + 00 | 2.019e + 01 | 3.494e − 03 | 4.6798 |
| 0.0500 | 7390 | 10 | 1.452e + 00 | 1.023e + 01 | 1.168e + 02 | 1.363e − 04 | 3.9690 |
| 0.0250 | 26,956 | 8 | 5.592e + 00 | 8.501e + 01 | 7.248e + 02 | 8.704e − 06 | 2.4746 |
| 0.0125 | 103,209 | 8 | 2.027e + 01 | 5.367e + 02 | 4.456e + 03 | 1.566e − 06 | |

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 rate of convergence estimated from 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 Fig. 6(b)). The exact solution we choose is

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

where $S$ is the Stokeslet, $R$ is the Rotlet, $\alpha_m$ and $\beta_m$ are unit vectors with random direction, and $p_m$ are points in the spheres different from the points $z_m$ used in the integral formulation. The test results are shown in Table 4.

**Example 3.** We solve the Navier equation on a single-surface bounded domain. The boundary surface (Fig. 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.

**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 (Fig. 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 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.

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

| $h$ | $N$ | Iterations | FMM(s) | Local(s) | Total(s) | Error | Rate |
|---|---|---|---|---|---|---|---|
| 0.1000 | 3072 | 22 | 2.050e + 00 | 1.070e + 00 | 6.864e + 01 | 1.408e − 03 | 3.2230 |
| 0.0500 | 12,288 | 21 | 1.080e + 01 | 8.250e + 00 | 4.001e + 02 | 1.508e − 04 | 5.6594 |
| 0.0250 | 43,200 | 23 | 6.927e + 01 | 6.566e + 01 | 3.103e + 03 | 2.984e − 06 | 3.5413 |
| 0.0125 | 172,800 | 20 | 3.456e + 02 | 6.279e + 02 | 1.947e + 04 | 2.563e − 07 | |

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

| $h$ | $N$ | Iterations | FMM(s) | Local(s) | Total(s) | Error | Rate |
|---|---|---|---|---|---|---|---|
| 0.1000 | 2320 | 13 | 1.170e + 00 | 1.840e + 00 | 3.913e + 01 | 9.822e − 04 | 4.8379 |
| 0.0500 | 8504 | 12 | 7.920e + 00 | 1.234e + 01 | 2.431e + 02 | 3.435e − 05 | 3.4423 |
| 0.0250 | 31,528 | 12 | 4.337e + 01 | 1.123e + 02 | 1.868e + 03 | 3.160e − 06 | 3.1018 |
| 0.0125 | 122,736 | 12 | 1.602e + 02 | 8.543e + 02 | 1.2904+04 | 3.681e − 07 | |

Table 6
Example 4. Poisson equation on a single-surface bounded domain (Fig. 6(c))

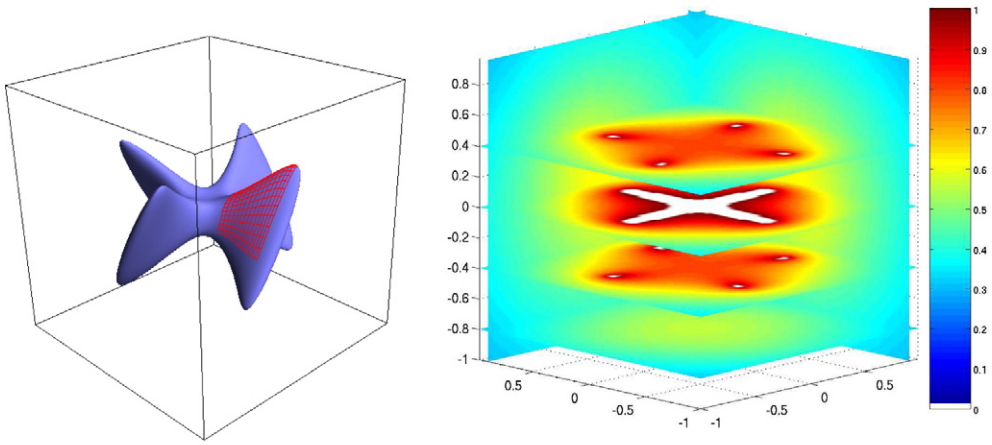
| $h$ | $N$ | Iterations | FMM(s) | Local(s) | Total(s) | Error | Rate |
|---|---|---|---|---|---|---|---|
| 0.1000 | 1688 | 10 | 1.700e − 01 | 1.250e + 00 | 1.420e + 01 | 3.166e − 03 | 3.6949 |
| 0.0500 | 5480 | 10 | 1.140e + 00 | 7.460e + 00 | 8.600e + 01 | 2.445e − 04 | 4.0170 |
| 0.0250 | 20,736 | 10 | 5.120e + 00 | 6.346e + 01 | 6.858e + 02 | 1.510e − 05 | 2.9193 |
| 0.0125 | 78,792 | 10 | 1.883e + 01 | 4.554e + 02 | 4.742e + 03 | 1.996e − 06 | |

Only the data for the boundary integral solver are shown.

**Example 5.** This example demonstrates the handling of more complex geometry. We solve the Laplace problem on an unbounded domain with boundary shown in Fig. 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 reported in Table 7. Fig. 7(b) shows the potential field on several slices of the domain.

**Example 6.** This example computes the Stokes flow around a static object (Fig. 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 $(\pm 1, \pm 1, \pm 2)$. Fig. 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.

**Example 7.** Here, we compute the Stokes flow around a shape similar to starfish vesicles [48], Fig. 9(a). The boundary of the domain is shown in Fig. 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 $(\pm 1, \pm 1, \pm 2)$. Fig. 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.




(a) (b)

Fig. 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.

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

| $h$ | $N$ | Iterations | 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 | 29,144 | 21 | 6.510e + 00 | 8.822e + 01 | 1.989e + 03 |

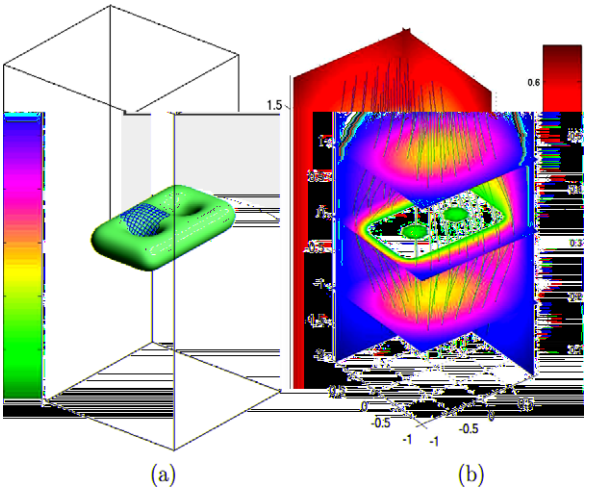



Fig. 8. Example 6. Stokes equations on an unbounded domain. (a) The boundary surface of the domain and the rectangle region where the solution is evaluated. (b) Velocity magnitude for several cross-sections and streamlines of the fluid field.

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

| $h$ | $N$ | Iterations | FMM(s) | Local(s) | Total(s) |
|---|---|---|---|---|---|
| 0.0500 | 5465 | 84 | 3.750e + 00 | 6.600e + 00 | 8.694e + 02 |
| 0.0250 | 19,684 | 69 | 2.031e + 01 | 5.545e + 01 | 5.227e + 03 |

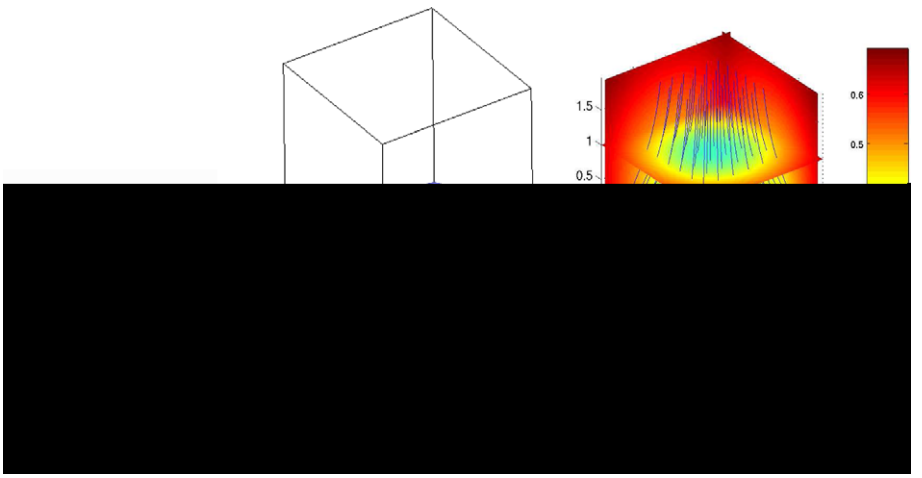



Fig. 9. Example 7. Stokes equations with an unbounded domain. (a) A starfish vesicle from [48]. (b) The boundary surface of the domain. (c) The velocity magnitude and flow streamlines.

In the first four examples which have exact solutions, we observe an overall convergence rate between 3 and 4. In each example, the time of FMM computation increases linearly with respect to the number of the quadrature points. On the other hand, the time of performing the local corrections for integration near singularities grows more rapidly: it increases by a factor of eight as the number of Nyström points quadruples. These results match the analysis in Section 3.2.

Table 9
Example 7. Stokes equations with an unbounded domain

| $h$ | $N$ | Iterations | FMM(s) | Local(s) | Total(s) |
|---|---|---|---|---|---|
| 0.0500 | 4304 | 120 | 2.955e + 00 | 5.745e + 00 | 1.044e + 03 |
| 0.0250 | 16,636 | 91 | 1.823e + 01 | 5.031e + 01 | 6.237e + 03 |

## 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 [50], other components (local corrections for singular integration, nearly singular integration) are not. Although they should be relatively easy to parallelize due to the 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. Similarly, performance of many parts of the code can be improved.

Finally, we have only considered domains with smooth boundaries in this paper. However, domains with sharp features play an important role in many engineering applications. Extension of the current approach to domains with piecewise smooth boundaries is another important research direction, which likewise is no trivial if we want to preserve the high-order convergence rate.

### 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 thank Leslie Greengard and Michael Shelley for numerous discussions and suggestions. We thank anonymous referees for their comments.

## Appendix 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

$$\int_U D(\boldsymbol{x}, g(c))\boldsymbol{\psi}(c)\,\mathrm{d}c \quad \text{and} \quad \int_U K(\boldsymbol{x}, g(c))(\boldsymbol{\psi}(c) - \boldsymbol{\psi}(c'))\,\mathrm{d}c \tag{26}$$

over planar domains $U$ for $\boldsymbol{x} = g(c')$ for a fixed $c' \in U$. We assume that the values at quadrature points are computed exactly, $g(\cdot)$ is $C^\infty$-continuous but not necessarily analytic, its Jacobian is of maximal rank and the functions $\boldsymbol{\phi}$ are $C^M$ continuous for some $M \geqslant 3$. The assumption on continuity of $g(\cdot)$ can be relaxed if necessary, but we use the stronger assumption (of $C^\infty$ smoothness) as all the surface parameterizations we use are $C^\infty$-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

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

Let $I_z$ be the operator $f \to \int_0^{t_z} f(z,\ldots)\,\mathrm{d}z$, 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 $z$: $I_z^h f = \sum_{i=0}^{N} h f(z_i,\ldots)$. 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:

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

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 $\partial_u$ commutes with $I_v$ and $\partial_v$ commutes with $I_u^{h_u}$, we obtain the estimate

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

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

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

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

**Lemma A.1.** *Suppose $f(q) : A \to \mathbb{R}$, 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 $m \leqslant M-1$ 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 $\rho \to 0$, for $i + j \leqslant M - k$, and the limit is a continuous function of $\theta$. To simplify exposition, we consider $\partial_\rho^j$ only; the general case is analogous.

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

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

where $R^{m+1}\rho^{-m} \to 0$ for $\rho \to 0$. 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 $\partial_\rho^j f$:

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

where $R_j^{m+1-j}\rho^{-(m-j)} \to 0$ for $\rho \to 0$. 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 $\rho$; the limit at $\rho = 0$ can be easily verified 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 $R_l^{m+1-l}\rho^{-(m-l)} \to 0$ as $\rho \to 0$ (it is a remainder term in Taylor expansion of $\partial_\rho^l f$ of degree $m-l$), the whole sum converges to zero if $m \geqslant j + k$. 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 $k + j \leqslant M$. 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 $\theta$ and $\rho$ are bounded, in the expressions for the derivatives of $f$ the only factors that are not bounded have the form $\rho^{-l}$, and one can show by induction that for derivatives of order $m$, the maximal value of $l$ is $m$. □

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

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

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

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

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

As $g$ is non-singular at $c = c'$, all components of $r/\rho$ are bounded from below, as is $r^2/\rho^2$. For $(r\cdot n(g(c)))$, an explicit calculation using Taylor expansions shows that the linear terms in $\rho$ vanish, and the first non-zero terms are of order $\rho^2$.

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

As $r^2/\rho^2$ 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 $C^\infty$-continuous.

Using the second part of Lemma A.1, we conclude that the growth of $m$th derivatives of $\rho D$ is bounded by $\rho^m$. 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 $\rho^{m+1}$. □

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

**Proof.** We consider two parts of the integral separately.

*Non-singular 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 $\chi\colon[0,\infty)\to[0,1]$ is a $C^\infty$ function which is 1 in a neighborhood of zero, and $\chi(r) = 0$ for $r\geqslant 1$.

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 $\partial_j^M(D(g(c\prime),g(c))(1-\eta(c))\boldsymbol{\phi}(c))$, we obtain

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

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 $\theta$; 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 $\eta_0$ does not depend on $\theta$. Similarly to the non-singular case, we obtain estimates for $\partial_\rho^m(f\eta_0)$ by using the chain rule to bound derivatives of $\eta_0$ 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 $\partial_\theta^M(f\eta_0) = \mathrm{O}(1)$ and $\partial_\rho^M(f\eta_0) = \mathrm{O}(h^{-M/2})$, we obtain the error bound of order $\mathrm{O}(h^{M/2})$.

Combining the error bounds for the singular and non-singular parts yield the statement of the theorem. □

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

**Lemma A.4.** *Suppose that $f : \mathbb{R} \to \mathbb{R}$ 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 $\mathrm{O}(h^{\frac{M-1}{2}})$ error for the integral*

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

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

$$\int_{-\sqrt{h}}^{\sqrt{h}} \frac{1}{\rho} S(\rho)\chi\left(\frac{|\rho|}{\sqrt{h}}\right) \mathrm{d}\rho = \int_{-\sqrt{h}}^{\sqrt{h}} \frac{1}{\rho} (S(\rho) - S(0))\chi\left(\frac{|\rho|}{\sqrt{h}}\right) \mathrm{d}\rho + \left(\int_{-\sqrt{h}}^{\sqrt{h}} \frac{1}{\rho}\chi\left(\frac{|\rho|}{\sqrt{h}}\right) \mathrm{d}\rho\right) S(0).$$

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 $\mathrm{O}(h^{\frac{M-1}{2}})$ 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 $\mathrm{O}(h^{\frac{M-1}{2}})$ for the principal value of the integral. □

**Lemma A.5.** *Let $g(c)$ be a $C^\infty$-continuous regular parametrization of a part of the boundary $\Gamma$, $c'$ a fixed point in the parametrization domain $C$ containing a disk of radius $a$ centered at $c'$, and $(\rho,\theta)$ is a polar coordinate system centered at $c'$. Suppose $\phi$ is $C^M$.*

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

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

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

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

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

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

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 $K(g(c'), g(c(\rho,\theta)))(\phi(y) - \phi(x))$ are $\mathrm{O}(\rho^{-(m+3)})$. □

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

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

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 $K(g(c'),g(c))(\boldsymbol{\psi}(c)-\boldsymbol{\psi}(c'))$ 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 $K(g(c'),g(c))(1-\eta_{c'}(c))(\boldsymbol{\psi}(c)-\boldsymbol{\psi}(c'))$ 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:

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

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

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

has an error bound $O(h^{\frac{M-2}{2}})$. Since (29) as a function of $\theta$ 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 $O(h^{\frac{M-3}{2}})$. □

## Appendix B. Derivation of pressure jump

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

$$[[\boldsymbol{u}]]=\boldsymbol{\phi}, \tag{30}$$

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

$$[[\operatorname{div}\boldsymbol{u}]]=0. \tag{32}$$

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

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

This implies the following six equalities:

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

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

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

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

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

which is equivalent to

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

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

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

Putting it back into (31), we have

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

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

**Proof.** For a fixed $x$, the integral over a single chart $U$ is

$$\int_U D(\boldsymbol{x}, g(c))\boldsymbol{\psi}(c)\,\mathrm{d}c,$$

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 $\partial_j^M(D(\boldsymbol{x}, g(c))\boldsymbol{\psi}(c))$ for $j = 1, 2$. Since $g$ is $C^\infty$ and $\phi$ is $C^M$, all $M$th order derivatives of $\psi(c)$ are uniformly bounded and it suffices to control $\partial_j^M D(\boldsymbol{x}, g(c))$. Furthermore, all $M$th order derivatives of $D(x, g(c))$ with respect to the chart parameterization, are bounded by a multiple of supremum of all $M$th order derivatives of $D(x, y)$, with respect to $y$.

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

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

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

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

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

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

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

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

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

**Proof.** The limit velocity at $x_0$, $\frac{1}{2}\boldsymbol{\phi}(\boldsymbol{x}_0) + D\boldsymbol{\phi}(\boldsymbol{x}_0)$, has convergence rate $\mathrm{O}(h^{\frac{M}{2}-1})$ by Theorem A.3. From the

[26] N. Heuer, M.E. Mellado, E.P. Stephan, A *p*-adaptive algorithm for the BEM with the hypersingular operator on the plane screen, Int. J. Numer. Meth. Eng. 53 (1) (2002) 85–104, *p* and *hp* finite element methods: mathematics and engineering practice (St. Louis, MO, 2000) 0029-5981.
[27] J. Higdon, G. Muldowney, Resistance functions for spherical particles, droplets and bubbles in cylindrical tubes, J. Fluid Mech. 298 (1995) 193–210.
[28] T.Y. Hou, J.S. Lowengrub, M.J. Shelley, Boundary integral methods for multicomponent fluids and multiphase materials, J. Comput. Phys. 169 (2) (2001) 302–362, 0021-9991.
[29] Q. Huang, T.A. Cruse, Some notes on singular integral techniques in boundary element analysis, Int. J. Numer. Meth. Eng. 36 (15) (1993) 2643–2659, 0029-5981.
[30] P.R. Johnston, Application of sigmoidal transformations to weakly singular and near-singular boundary element integrals, Int. J. Numer. Meth. Eng. 45 (10) (1999) 1333–1348, 0029-5981.
[31] A. Khodakovsky, N. Litke, P. Schröder, Globally smooth parameterizations with low distortion, ACM Trans. Graph. 22 (3) (2003) 350–357.
[32] S. Kim, S.J. Karrila, Microhydrodynamics: Principles and Selected Applications, Butterworth-Heinemann, London, 1991.
[33] R. Kress, Linear integral equations, Applied Mathematical Sciences, Springer, Berlin, 1999.
[34] C. Lage, C. Schwab, Wavelet Galerkin algorithms for boundary integral equations, SIAM J. Sci. Comput. 20 (6) (1999) 1095–7197, 2195–2222 (electronic).
[35] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (2) (1984) 285–299.
[36] S.G. Mikhlin, Integral Equations and their Applications to Certain Problems in Mechanics, Mathematical Physics and Technology, second revised ed., Translated from the Russian by A.H. Armstrong, A Pergamon Press Book, The Macmillan Co., New York, 1964.
[37] G. Muldowney, J. Higdon, A spectral boundary element approach to three-dimensional Stokes flow, J. Fluid Mech. 298 (1995) 167–192.
[38] K. Nabors, J. Phillips, F.T. Korsmeyer, 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, SIAM, Philadelphia, PA, 1995, pp. 193–215.
[39] N. Nishimura, Fast multipole accelerated boundary integral methods, Appl. Mech. Rev. 55 (4) (2002) 234–299.
[40] H. Power, L. Wrobel, Boundary Integral Methods in Fluid Mechanics, Computational Mechanics Publications, 1995.
[41] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press, Cambridge, 1992.
[42] C. Pozrikidis, Overview of dynamical simulations of the flow of suspensions of liquid capsules and drops, in: Free Boundary Problems: Theory and Applications (Crete, 1997)Chapman & Hall/CRC Res. Notes Math., vol. 409, Chapman & Hall/CRC, Boca Raton, FL, 1999, pp. 82–88.
[43] C. Schwab, W.L. Wendland, Numerical integration of singular and hypersingular integrals in boundary element methods, in: Numerical Integration (Bergen, 1991)NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., 357, Kluwer Academic Publishers, Dordrecht, 1992, pp. 203–218.
[44] J. Tausch, Sparse BEM for potential theory and stokes flow using variable order wavelets, Comput. Mech. 32 (4–6) (2003) 312–318.
[45] J. Tausch, A variable order wavelet method for the sparse representation of layer potentials in the non-standard form, J. Numer. Math. 12 (3) (2004) 233–254.
[46] J. Tausch, J. White, Multiscale bases for the sparse representation of boundary integral operators on complex geometry, SIAM J. Sci. Comput. 24 (5) (2003), 1610–1629 (electronic).
[47] S. Vijayakumar, D.E. Cormack, A new concept in near-singular integral evaluation: the continuation approach, SIAM J. Appl. Math. 49 (5) (1989) 1285–1295.
[48] W. Wintz, H.-G. Döbereiner, U. Seifert, Starfish vesicles, Europhys. Lett. 33 (1996) 403–408.
[49] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, J. Comput. Phys. 196 (2) (2004) 591–626.
[50] L. Ying, G. Biros, D. Zorin, H. Langston, A new parallel kernel-independent fast multipole method, in: SC '03: Proceedings of the 2003 ACM/IEEE Conference on Supercomputing, IEEE Computer Society, Washington, DC, USA, 2003, p. 14.
[51] L. Ying, D. Zorin, A simple manifold-based construction of surfaces of arbitrary smoothness, ACM Trans. Graph. 23 (3) (2004) 271–275.
[52] A.Z. Zinchenko, R.H. Davis, An efficient algorithm for hydrodynamical interaction of many deformable drops, J. Comput. Phys. 157 (2) (2000) 539–587.