A Fast Integral Equation Method for the Two-Dimensional Navier-Stokes Equations

Ludvig af Klinteberg[a], Travis Askham[b], Mary Catherine Kropinski[a]

[a] *Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada.*
[b] *Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ, USA.*

Abstract

The integral equation approach to partial differential equations (PDEs) provides significant advantages in the numerical solution of the incompressible Navier-Stokes equations. In particular, the divergence-free condition and boundary conditions are handled naturally, and the ill-conditioning caused by high order terms in the PDE is preconditioned analytically. Despite these advantages, the adoption of integral equation methods has been slow due to a number of difficulties in their implementation. This work describes a complete integral equation-based flow solver that builds on recently developed methods for singular quadrature and the solution of PDEs on complex domains, in combination with several more well-established numerical methods. We apply this solver to flow problems on a number of geometries, both simple and challenging, studying its convergence properties and computational performance. This serves as a demonstration that it is now relatively straightforward to develop a robust, efficient, and flexible Navier-Stokes solver, using integral equation methods.

Keywords: Navier-Stokes equations, Integral equations, Function extension, Quadrature

1. Introduction

Fast integral equation methods (FIEMs) - techniques based on integral equations coupled with associated fast algorithms for the integral operators such as the Fast Multipole Method (FMM) [33] or fast direct solvers [49] - have become the tool of choice for solving the linear, elliptic boundary value problems associated with the Laplace and Helmholtz equations, the equations of elasticity, the Stokes equation, and other classical elliptic equations in mathematical physics. For these boundary value problems, FIEMs have been a game changer: superior stability and accuracy, efficiency through dimension reduction and acceleration, and ease of adaptivity are a few of the more significant advantages.

While nonlinear equations are not directly amenable to solution via integral equation methods, for some equations a suitable temporal discretization and/or linearization gives rise to a sequence of "building-block" partial differential equations (PDEs) of the form of the classic equations mentioned above. These can then be targeted with suitably-chosen FIEMs. The focus of this paper is to discuss a FIEM-based solver for the two-dimensional incompressible Navier-Stokes equations (INSE),

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} &= -\nabla p + \frac{1}{Re}\Delta \boldsymbol{u}, && \boldsymbol{x}\in\Omega,\\ \nabla\cdot\boldsymbol{u} &= 0, && \boldsymbol{x}\in\Omega,\\ \boldsymbol{u} &= \boldsymbol{f}, && \boldsymbol{x}\in\partial\Omega, \end{aligned} \tag{1}$$

on a general bounded domain Ω and satisfying the no-slip boundary conditions. In this context, the basic idea is as follows. Applying a semi-implicit temporal discretization to the momentum equation in (1) gives rise to a sequence of modified biharmonic (in the stream function formulation, c.f. [42]) or modified Stokes equations,

$$
\begin{aligned}
(\alpha^2 - \Delta)\boldsymbol{u}^{N+1} + Re\,\nabla p^{N+1} &= \boldsymbol{F}^N, && \boldsymbol{x} \in \Omega,\\
\nabla \cdot \boldsymbol{u}^{N+1} &= 0, && \boldsymbol{x} \in \Omega,\\
\boldsymbol{u}^{N+1} &= \boldsymbol{f}, && \boldsymbol{x} \in \partial\Omega,
\end{aligned}
\tag{2}
$$

where $\alpha^2 = O(Re\,/\delta t)$, δt is the time step, and $\boldsymbol{F}^N$ is a volume term that carries information from the previous time step. This is the so-called Rothe's method [19], which sometimes is referred to as the transverse method of lines, or the method of lines transpose [18]. It is (2), then, that are recast by representing the solution as the sum of suitably chosen layer and volume potentials. This representation will satisfy the incompressibility constraint by construction, thereby eliminating the need for artificial boundary conditions associated with projection and other methods based on direct discretization of the Navier-Stokes equations. The focus of this paper, then, is to present a FIEM for (2) in general, two-dimensional bounded domains and to examine their performance for a range of Re.

This general approach was examined in [31] for the special case of flow inside a circular cylinder. It was demonstrated in this paper that FIEMs have a great deal of potential computing solutions for Reynolds number up to $O(1000)$ with both high spatial and temporal accuracy. Despite inspiring some other integral equation-based solvers for the INSE [15], this work has yet to be fully realized in general 2-D or 3-D domains. As outlined in [45], there have been two critical stumbling blocks in achieving this goal. The first is how to evaluate the layer potentials close to the boundary where the kernels become nearly singular. There are many high-order quadrature methods that are well-suited for discretizing the integral operators and evaluating potentials away from the boundary [35], but these break down when the target becomes close to source points on the boundary. The second stumbling block is how to solve inhomogenous equations on general domains. A potential theoretic approach expresses the particular solution as a volume potential: a convolution of the fundamental solution with the right hand side of the equation. Unfortunately, available fast methods for evaluating layer potentials are generally in the form of "box codes" such as [21] or the FFT. These require knowing the source data throughout a square or cube. In order for these box codes to be effective, this data must be of sufficient regularity throughout the box. For a boundary value problem on a general domain, this then requires extending the right-hand side with the required regularity throughout the bounding box.

Recently, significant progress has been made on both the quadrature and function extension fronts. Developing specialized quadrature that can be used to evaluate potentials arbitrarily close to the boundary, and which can be efficiently combined with fast summation algorithms, has been an area of active research. One approach is "Quadrature By Expansion", or QBX [11, 44], which uses local expansions of the solution with expansion centers close to the boundary. QBX has the advantage that it can be extended to three dimensions, but in order to be efficient it needs to be integrated within an FMM on a fundamental algorithmic level. This is nontrivial, although recent work has demonstrated that it is feasible [61, 62]. Another approach is an explicit kernel-split, panel-based Nyström method outlined in [37, 39]. Here, the kernel is split into its smooth and singular parts, with the smooth parts being evaluated using a composite n-point Gaussian quadrature scheme, and the singular parts using specialized quadrature based on product integration. An example of a recent implementation of this scheme can be found in [64], in the context of Stokes flow in complex domains with corners. In our case, the kernel-splitting of the modified Stokes kernel is considerably more complex, with an additional complication that the underlying kernels require more spatial resolution for larger values of α (corresponding to large Re or small time steps). A straightforward implementation of [37] shows a breakdown in accuracy which requires careful local refinement to alleviate this [6]. The result is a quadrature scheme which is robust for arbitrary target points and values of α.

Two approaches have been investigated recently which extend functions throughout a bounding box in which the geometry of the problem is embedded. One approach uses extension via layer potentials [8, 10]. The layer potential is determined by enforcing continuity across the boundary of the domain through solving an integral equation, and it is then used to construct the function extension in the complement of the domain. Requiring a higher degree of continuity means the integral equation is based on solving a higher-order partial differential equation and this can get prohibitively complex. A second approach, which is the approach we take here, is to smoothly extend the right-hand-side of (4) via extrapolation using radial basis functions [26, 27]. This method is called Partition of Unity eXtension, or PUX. It is then simple to convolve this extended function with the periodic Green's function for these equations using a fast Fourier transform (FFT) and the solution may be evaluated on the boundary rapidly with a non-uniform FFT [12].

The methods we propose in this paper consist of coupling together existing and custom-developed state-of-the-art tools for constructing and evaluating the layer and volume potentials. The key elements of the overall approach and the road map to where these elements are discussed in this paper are summarized below:

1. Using PUX, the function $\boldsymbol{F}$ is extended onto a uniform grid discretizing the bounding box. The volume potential is evaluated on this grid via an FFT and evaluated on $\partial\Omega$ via a non-uniform FFT. This is discussed in section 3.1.

2. To the volume potential, layer potentials are added which ensure the solution satisfies the no-slip boundary conditions $\boldsymbol{u} = \boldsymbol{f}$. This involves solving a system of Fredholm integral equations of the second kind for the homogenous modified Stokes equations (see section 2.2). After applying kernel-split quadrature [39, 37, 6], outlined in section 4, the resulting linear system is dense and of the form of $[I + K]\boldsymbol{\mu} = \mathbf{b}^N$, where N represents the time step and K has low-rank structures. Since the boundary $\partial\Omega$ is fixed, the system matrix $[I + K]$ does not change throughout time. We use a fast direct method [28, 48] which exploits the low-rank off-diagonal blocks to invert the system matrix as a precomputation. This inverse can be applied to the updated vector $\mathbf{b}^N$ at each time step at a fraction of the time. All computations are linear in computational time. This is discussed in section 3.2.2.

3. Given the density of the layer potentials, $\boldsymbol{\mu}$, from the previous step, $\boldsymbol{u}$ and $\nabla\boldsymbol{u}$ are evaluated at all regular grid points in Ω. Kernel-split quadrature is used here, too, when evaluating the layer potentials close to the source terms. These point-to-point interactions are accelerated by an FMM [33] based on the separation of variables formulae derived in [9], which are stable for all values of α (section 3.2.1). These values are used to construct a new right hand side $\boldsymbol{F}$ in Ω.

4. The above steps are repeated for the next time step. The solution is advanced using a semi-implicit spectral deferred correction method [50]. This is discussed in section 3.3.

The result is a Navier-Stokes solver which has linear or near-linear scaling in computational time at each time step; specifically the computational cost is $\mathcal{O}(N_\Omega \log N_\Omega + N_\Omega + N_{\partial\Omega})$, where N_Ω and $N_{\partial\Omega}$ denote the number of discretization nodes in the domain and on the boundary, respectively. We show in section 5.5 that the stability requirement is $\delta t = O(1/\mathit{Re})$. The numerical experiments, which we present in section 5, indicate that our methods are well suited for problems with moderate values of the Reynolds number (we consider examples up to $\mathit{Re} = 200$).

Our approach could be described as an embedded boundary approach, which is a popular approach for methods based on direct discretization of the PDE, such as the Immersed Boundary (IB) method of Peskin [53]. Here, the effect of the boundary is smeared to neighboring grid cells (including the cut cells) by using approximations to the Dirac δ function. The original IB framework was limited to low order accuracy; recently, Stein et al. [59] presented a higher-order IB approach based on envisioning a smooth extension of the solution outside of the original domain. The ideas behind the resulting method, IB Smooth Extension (IBSE), bear some resemblance to the present work but are based on introducing new unknowns for the solution outside of the domain as opposed to extending the known inhomogeneity, as we describe below. Further references for embedded boundary methods in the PDE literature can be found in [59].

There has been recent work on time-dependent integral equation methods for the unsteady Stokes equations [30, 34, 63], which could form the basis of a FIEM-based Navier-Stokes solver quite different from the one presented here. These methods are still in the early stages of development, but are nevertheless promising. In particular, their formulation allows for a natural treatment of moving boundaries, and would likely have advantages in terms of stability.

The remainder of this paper is organized as follows. Section 2 introduces the underlying mathematical formulation. Section 3 outlines the framework of numerical methods. Section 4 gives a detailed description of the quadrature used for singular and nearly singular integrals. Section 5 reports on results from numerical experiments. Finally, concluding remarks are provided in section 6, while supplementary material is available in appendices A–D.

2. Formulation

Many numerical approaches to solving the Navier-Stokes equations begin by performing a discretization of the equations in time. Because it is simpler to handle the nonlinear term explicitly but less restrictive to handle the stiff diffusion term implicitly, it is common to use an implicit-explicit (IMEX) method [7]. Using the first-order IMEX-Euler rule results in the time-independent elliptic PDE

$$\begin{aligned} \frac{\boldsymbol{u}^{N+1}-\boldsymbol{u}^{N}}{\delta t} - \frac{1}{Re}\Delta \boldsymbol{u}^{N+1} + \nabla p^{N+1} &= -(\boldsymbol{u}^N\cdot\nabla)\boldsymbol{u}^N, && \boldsymbol{x}\in\Omega,\\ \nabla\cdot\boldsymbol{u}^{N+1} &= 0, && \boldsymbol{x}\in\Omega,\\ \boldsymbol{u}^{N+1} &= \boldsymbol{f}, && \boldsymbol{x}\in\partial\Omega. \end{aligned} \tag{3}$$

The higher-order IMEX rules result in qualitatively similar elliptic PDEs. Further, the spectral deferred corrections (SDC) method [22] can be used to combine many low order IMEX steps into a high order scheme for advection-diffusion problems [50], which will be the approach of this paper.

It is in the solution of the elliptic PDE (3), called the modified Stokes equations[1] where the PDE discretization and integral equation methods diverge. Because the equation is linear in the unknowns $\boldsymbol{u}^{N+1}$, the solution of (3) can be split into a particular and a homogeneous problem,

$$\begin{aligned} (\alpha^2-\Delta)\boldsymbol{u}^P + Re\,\nabla p^P &= \boldsymbol{F}, && \boldsymbol{x}\in\Omega,\\ \nabla\cdot\boldsymbol{u}^P &= 0, && \boldsymbol{x}\in\Omega, \end{aligned} \tag{4}$$

and

$$\begin{aligned} (\alpha^2-\Delta)\boldsymbol{u}^H + Re\,\nabla p^H &= 0, && \boldsymbol{x}\in\Omega,\\ \nabla\cdot\boldsymbol{u}^H &= 0, && \boldsymbol{x}\in\Omega,\\ \boldsymbol{u}^H &= \boldsymbol{f}-\boldsymbol{u}^P, && \boldsymbol{x}\in\partial\Omega\ , \end{aligned} \tag{5}$$

where

$$\alpha^2 = \frac{Re}{\delta t}, \qquad \text{and} \qquad \boldsymbol{F} = \alpha^2\boldsymbol{u}^N - Re(\boldsymbol{u}^N\cdot\nabla)\boldsymbol{u}^N. \tag{6}$$

The forced PDE (4) has no boundary conditions, so that a candidate particular solution may be computed by convolving the right-hand-side of (4) with a free-space or periodic Green's function — in particular, a Green's function which is independent of the domain and analytically known. Regardless of the particular solution chosen, the homogeneous equation (5) is specified so that the composite solution,

$$\boldsymbol{u}^{N+1} = \boldsymbol{u}^P + \boldsymbol{u}^H, \tag{7}$$

$$p^{N+1} = p^P + p^H, \tag{8}$$

satisfies the original problem (3). For this reason, the homogeneous solution is sometimes referred to as a boundary correction.

In an integral equation method, the homogeneous solution is generally represented by the convolution of an unknown density defined on the boundary, $\partial\Omega$, with an integral kernel given in terms of the equivalent of a charge or a dipole of the equations (5), which a priori satisfies the divergence-free condition and the homogeneous equations inside the domain. For a well-chosen representation, the enforcement of the boundary

[1] These equations may also be referred to as the linearized unsteady Stokes equations [54], or the Brinkman equations [17].

conditions gives a well-conditioned equation for the density on the boundary. Once this boundary integral equation is solved, the homogeneous solution may be evaluated at any point in the domain by convolution.

The central equation of our method is the inhomogeneous modified Stokes equations with Dirichlet boundary conditions,

$$\begin{aligned} (\alpha^2 - \Delta)\boldsymbol{u} + \nabla p &= \boldsymbol{F}, && \boldsymbol{x} \in \Omega, \\ \nabla \cdot \boldsymbol{u} &= 0, && \boldsymbol{x} \in \Omega, \\ \boldsymbol{u} &= \boldsymbol{f}, && \boldsymbol{x} \in \partial\Omega. \end{aligned} \tag{9}$$

This stationary problem is the result of a semi-implicit discretization of the Navier-Stokes equations (3). Because of the divergence-free condition, these equations, like the Navier-Stokes equations, imply a compatibility condition on $\boldsymbol{f}$, namely

$$\int_{\partial\Omega} \boldsymbol{f} \cdot \hat{\boldsymbol{n}} \, \mathrm{d}S_{\boldsymbol{y}} = 0 \, . \tag{10}$$

In analogy with the regular Stokes equations, we call the free-space Green's function of these equations the modified stokeslet tensor, or simply the stokeslet. It is defined as

$$S(\boldsymbol{x}, \boldsymbol{y}) = (\nabla \otimes \nabla - \Delta) G(\boldsymbol{x}, \boldsymbol{y}), \tag{11}$$

where G is the free-space Green's function of the modified biharmonic equation [42], i.e. $\Delta(\Delta - \alpha^2)G = \delta$,

$$G(\boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{2\pi\alpha^2} \left(\log \|\boldsymbol{x} - \boldsymbol{y}\| + K_0(\alpha \|\boldsymbol{x} - \boldsymbol{y}\|) \right) . \tag{12}$$

Here K_0 is the zeroth modified Bessel function of the second kind. The corresponding pressure vector $\boldsymbol{\phi}$ and stress tensor T (which we will refer to as the stresslet) are defined as

$$\phi_i(\boldsymbol{x}, \boldsymbol{y}) = (\Delta - \alpha^2) \frac{\partial G(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i} \, , \tag{13}$$

$$T_{ijk}(\boldsymbol{x}, \boldsymbol{y}) = -\delta_{ik} \phi_j(\boldsymbol{x}, \boldsymbol{y}) + \frac{\partial S_{ij}(\boldsymbol{x}, \boldsymbol{y})}{\partial x_k} + \frac{\partial S_{kj}(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i} \, . \tag{14}$$

Note that if $\boldsymbol{u} = S\boldsymbol{\mu}$ is the field induced by a stokeslet with charge $\boldsymbol{\mu}$, then the stress tensor for that field, $\boldsymbol{\sigma}$, is given by

$$\sigma_{ik} = T_{ijk} \mu_j . \tag{15}$$

Closed-form expressions for S, $\boldsymbol{\phi}$, and T are derived in appendix B.

Our solution strategy for the equations (9) is based on two fundamental principles, which are typical of integral equation methods. The first principle is the use of potential theory, in which the solution is represented using convolutions with Green's functions of the equations. By construction, this guarantees that the divergence-free condition is satisfied, in contrast with methods based on direct discretization of the PDE. The second principle is the use of a composite solution, which is formed by the solutions to the particular problem (4) and the homogeneous problem (5). This split allows us to use numerical methods that are fast and geometrically flexible, as we shall see in section 3. In the remainder of this section, we describe in detail how the Green's functions are used to solve the particular and homogeneous problems.

2.1. Particular solution

From a mathematical point of view, the most straightforward solution to the particular problem (4) is the volume potential,

$$\boldsymbol{u}^P(\boldsymbol{x}) = \int_{\Omega} S_{ij}(\boldsymbol{x}, \boldsymbol{y}) \boldsymbol{F}_j(\boldsymbol{y}) \, \mathrm{d}V_{\boldsymbol{y}}, \tag{16}$$

which solves (4) using free-space boundary conditions. This is however not practical, as evaluating the integral (16) accurately over a complex domain would present a significant numerical challenge. Instead, we shall for the moment assume that Ω is embedded in a square domain $B = [0, 2\pi]^2$ (see fig. 1), and that there exists a compactly supported function $\boldsymbol{F}^e$ that is a q-continuous extension of $\boldsymbol{F}$ out into B,

$$\boldsymbol{F}^e = \boldsymbol{F} \quad \text{in } \Omega, \tag{17}$$

$$\Omega \subset \operatorname{supp}\{\boldsymbol{F}^e\} \subset B, \tag{18}$$

$$\boldsymbol{F}^e \in C^q(B), \quad q > 0. \tag{19}$$

We will return to the computation of the extension $\boldsymbol{F}^e$ in section 3.1.1. Having $\boldsymbol{F}^e$ lets us compute $\boldsymbol{u}^P$ as the solution to the extended particular problem,

$$\begin{aligned} (\alpha^2 - \Delta)\boldsymbol{u}^P + Re\, \nabla p^P &= \boldsymbol{F}^e, \qquad && \boldsymbol{x} \in B, \\ \nabla \cdot \boldsymbol{u}^P &= 0, && \boldsymbol{x} \in B. \end{aligned} \tag{20}$$

Note that the solution to the extended particular problem (20) also satisfies the original particular problem (4). Now, since B is a simple domain (a box), it would be straightforward to compute the particular solution as the volume potential over B with the extended function $\boldsymbol{F}^e$. Alternatively, and this is the path we will follow, we can compute it using a Fourier series. This amounts to solving (20) using periodic boundary conditions, which we can do since $\boldsymbol{F}^e$ is compactly supported, and therefore also periodic. Let

$$u_j^P(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^2} \hat{u}_j(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}}, \qquad p^P(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^2} \hat{p}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}}, \qquad F_j^e(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^2} \hat{F}_j^e(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}}. \tag{21}$$

Replacement into (20) and matching of coefficients gives the linear system of equations

$$(\alpha^2 + |\boldsymbol{k}|^2)\hat{u}_j(\boldsymbol{k}) + Re\, ik_j \hat{p}(\boldsymbol{k}) = \hat{F}_j^e(\boldsymbol{k}), \tag{22}$$

$$ik_j \hat{u}_j(\boldsymbol{k}) = 0, \tag{23}$$

with solution

$$\begin{aligned} \hat{u}_j(\boldsymbol{k}) &= \frac{|\boldsymbol{k}|^2 \delta_{jl} - k_j k_l}{|\boldsymbol{k}|^2(\alpha^2 + |\boldsymbol{k}|^2)} \hat{F}_l^e(\boldsymbol{k}), \quad \boldsymbol{k} \neq 0, \\ \hat{u}_j(0) &= \frac{1}{\alpha^2} \hat{F}_j^e(0). \end{aligned} \tag{24}$$

To rephrase the above, given a smooth extension of $\boldsymbol{F}$ with Fourier coefficients $\hat{\boldsymbol{F}}^e$, a solution of the particular equation, (4), is given by the convolution

$$u_j^P(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^2} \hat{S}_{jl}(\boldsymbol{k}) \hat{F}_l^e(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}}, \quad \boldsymbol{x} \in \Omega, \tag{25}$$

where $\hat{S}$ is the Fourier multiplier corresponding to the periodic stokeslet, i.e.

$$\begin{aligned} \hat{S}_{jl}(\boldsymbol{k}) &= \frac{\delta_{jl} - \hat{k}_j \hat{k}_l}{\alpha^2 + |\boldsymbol{k}|^2}, \quad \boldsymbol{k} \neq 0, \quad \hat{\boldsymbol{k}} = \boldsymbol{k}/|\boldsymbol{k}|, \\ \hat{S}_{jl}(0) &= \frac{\delta_{jl}}{\alpha^2}. \end{aligned} \tag{26}$$

We also need the gradient of the solution, in order to compute the right hand side of the next time step (6). This is straightforward to compute using Fourier differentiation,

$$\frac{\partial u_j^P(\boldsymbol{x})}{\partial x_m} = \sum_{\boldsymbol{k}\in\mathbb{Z}^2} ik_m \hat{S}_{jl}(\boldsymbol{k}) \hat{F}_l^e(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}}, \quad \boldsymbol{x} \in \Omega. \tag{27}$$

This solution procedure has some nice properties. First, once the extension $\boldsymbol{F}^e$ is available, computing the particular solution does not require that a linear system be solved; instead, the computation is based on evaluating a formula for the solution. Further, for smaller values of α, the multiplier $\hat{S}$ is strongly smoothing, i.e. it decreases the magnitude of higher Fourier modes. This property is both consistent with the physics and stabilizing from a numerical point of view. Unfortunately, this smoothing property is reduced for large values of α, corresponding to high Reynolds number flow.

Remark 1 (Small α)**.** *For $\alpha \ll 1$, the Fourier kernel* (26) *varies in magnitude by a factor $\mathcal{O}(\alpha^2)$ between $|\boldsymbol{k}| = 0$ and $|\boldsymbol{k}| = \mathcal{O}(1)$. We have observed that this leads to roundoff errors in the final solution ($\boldsymbol{u}^H + \boldsymbol{u}^P$) when $\boldsymbol{u}^p$ is computed numerically. However, we have found that these errors can be mitigated by artificially setting $\hat{S}(0) = 0$, such that $\boldsymbol{u}^p$ gets an error corresponding to the constant term of the Fourier series. Since a constant field is a solution to the homogeneous problem* (5)*, this discrepancy gets incorporated into $\boldsymbol{u}^H$ through the Dirichlet boundary condition, such that the final solution satisfies the inhomogeneous problem* (9)*.*

2.2. Homogeneous solution

The solution to the homogeneous problem (5) can be represented by a double layer potential, which we denote by

$$\boldsymbol{u}^H(\boldsymbol{x}) = \mathcal{D}[\boldsymbol{\mu}](\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \tag{28}$$

For a given double layer density $\boldsymbol{\mu}$ defined on $\partial\Omega$, the potential is defined as the boundary integral

$$\mathcal{D}_i[\boldsymbol{\mu}](\boldsymbol{x}) = \int_{\partial\Omega} D_{ij}(\boldsymbol{x}, \boldsymbol{y}) \mu_j(\boldsymbol{y}) \, \mathrm{d}S_{\boldsymbol{y}}, \tag{29}$$

where the double layer kernel, D, is defined using the stresslet (14) and the surface unit normal $\hat{\boldsymbol{n}}(\boldsymbol{y})$ pointing into Ω,

$$D_{ij}(\boldsymbol{x}, \boldsymbol{y}) = -T_{jik}(\boldsymbol{x}, \boldsymbol{y}) \hat{n}_k(\boldsymbol{y}). \tag{30}$$

By construction, this layer potential is divergence-free and satisfies the homogeneous modified Stokes equation in Ω.

We will now briefly describe how a double layer potential is used to solve the homogeneous Dirichlet problem. Much of the following depends on a number of technical results concerning the double layer potential which can be found in the literature [43, 55, 15] and are included in Appendix A for convenience. In contrast with the Stokes equation, we note that all of the following holds without modification for multiply-connected Ω. The appropriate density $\boldsymbol{\mu}$ is determined by enforcing the Dirichlet boundary condition $\boldsymbol{u}^H = \boldsymbol{f} - \boldsymbol{u}^P$ in the limit $\Omega \ni \boldsymbol{x} \to \partial\Omega$. Using the jump conditions for modified Stokes layer potentials (see Lemma 2 in appendix A), we obtain a second-kind integral equation in $\boldsymbol{\mu}$,

$$\frac{1}{2}\boldsymbol{\mu} + \mathcal{D}[\boldsymbol{\mu}] = \boldsymbol{f} - \boldsymbol{u}^P, \quad \boldsymbol{x} \in \partial\Omega. \tag{31}$$

This integral equation has a null space of rank 1 (see Lemma 3 in appendix A), so in order to have a unique solution we add the term

$$\mathcal{W}[\boldsymbol{\mu}](\boldsymbol{x}) = \frac{\hat{\boldsymbol{n}}(\boldsymbol{x})}{\int_{\partial\Omega} \mathrm{d}S} \int_{\partial\Omega} \boldsymbol{\mu}(\boldsymbol{y}) \cdot \hat{\boldsymbol{n}}(\boldsymbol{y}) \, \mathrm{d}S_{\boldsymbol{y}} \ , \tag{32}$$

which is sometimes referred to as a nullspace correction. Then, the homogeneous solution is given by a double layer potential (28), with the density satisfying

$$\frac{1}{2}\boldsymbol{\mu} + \mathcal{D}[\boldsymbol{\mu}] + \mathcal{W}[\boldsymbol{\mu}] = \boldsymbol{f} - \boldsymbol{u}^P, \quad \boldsymbol{x} \in \partial\Omega. \tag{33}$$

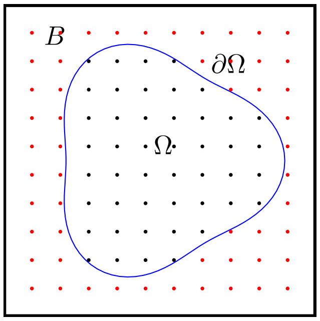


Figure 1: An illustration of the embedded boundary approach. The boundary of the domain is "floating" on top of a uniform grid, defined on a bounding box B. The forcing function $\boldsymbol{F}$, initially only known at grid points inside Ω (marked black), is smoothly extended to the function $\boldsymbol{F}^e$, which is defined at all grid points in B.

The new equation, (33), is invertible for any right hand side and the term $\mathcal{W}[\boldsymbol{\mu}]$ is zero provided that the right hand side satisfies the condition (see Lemma 4 in appendix A)

$$\int_{\partial\Omega} (\boldsymbol{f} - \boldsymbol{u}^p) \cdot \hat{\boldsymbol{n}} \, \mathrm{d}S_{\boldsymbol{y}} = 0 \, . \tag{34}$$

This automatically holds for $\boldsymbol{f}$ satisfying the compatibility condition, (10), and any particular solution, $\boldsymbol{u}^p$. Once this equation is solved, $\boldsymbol{u}^H$ can be evaluated anywhere inside Ω using (29).

3. Numerical methods

This section gives a complete outline of the numerical methods used. Quadrature will only be briefly discussed in this section, and more thoroughly covered in section 4.

3.1. Evaluating the particular solution

3.1.1. Smooth function extensions

In order to compute the particular solution $\boldsymbol{u}^P$ using the Fourier method of section 2.1, we need an extended forcing function $\boldsymbol{F}^e$ with a high order of regularity. To compute such a function extension, we use the method known as PUX, for Partition of Unity eXtension, introduced in [26]. The method is based on overlapping circular partitions with their centers distributed along the boundary $\partial\Omega$. Inside each partition radial basis functions (RBFs) are defined, which interpolate $\boldsymbol{F}$ at each point of the partition inside Ω, giving a local extrapolation of $\boldsymbol{F}$ to the other points in the partition. These exterior values in adjacent partitions, which can be multiply-defined due to the overlap, are then blended using a high-order partition of unity (POU) function, forming a function extension with high regularity. Finally, the extension is given compact support by blending it with a second layer of "zero partitions", located just outside Ω.

The current version of PUX is limited to a uniform grid in B, and we will use a uniform grid for that reason. Given a grid and a boundary, [26] provides heuristics for determining the parameters of the method; the only free parameters are the RBF shape parameter ϵ (which is set to 2 throughout) and the partition radius R. The accuracy largely depends on the number of grid points spanned by R; the more grid points spanned, the better the decay of the extension can be resolved, up to a limit. The range of points spanned by R is typically between 20 and 50.

3.1.2. Evaluating the volume potential

After applying the PUX algorithm, we have the values of the smooth extension $\boldsymbol{F}^e$ on a uniform grid in B. It is then straightforward to compute the particular solution $\boldsymbol{u}^P$ (25) in a fast way, using fast Fourier transforms (FFTs). First, two FFTs give us the Fourier coefficients $\hat{\boldsymbol{F}}^e(\boldsymbol{k})$. We can then compute the solution using two inverse FFTs for the volume grid, and two non-uniform FFTs of type 2 [12] for the discretization points on $\partial\Omega$. The gradient (27) is then computed using four additional inverse FFTs. Since the gradient is only needed on the volume grid, it does not require any non-uniform FFTs.

For a discretization with N_B uniform points in B, and $N_{\partial\Omega}$ nonuniform points on $\partial\Omega$, the computational cost is $\mathcal{O}(N_B \log N_B + N_{\partial\Omega})$. The method has spectral accuracy if $\boldsymbol{F}$ and its extension are smooth. If they are not smooth, the order of accuracy is related to the regularity of the extended function.

3.2. Computing the homogeneous correction

In order to solve the integral equation (33), we first need to discretize the boundary $\partial\Omega$ using a suitable quadrature. For this, we use a composite Gauss-Legendre quadrature, wherein $\partial\Omega$ is subdivided into N_p panels Γ_i of equal arclength,

$$\partial\Omega = \bigcup_{i=1}^{N_p} \Gamma_i, \tag{35}$$

each of which is discretized using an n-point Gauss-Legendre quadrature (we use $n = 16$ throughout this work). If we let $\boldsymbol{y}_i(t)$ be the mapping from the standard interval $[-1, 1]$ to the panel Γ_i, then

$$\int_\Gamma f(\boldsymbol{y})\,\mathrm{d}S_{\boldsymbol{y}} = \sum_{i=1}^{N_p} \int_{\Gamma_i} f(\boldsymbol{y})\,\mathrm{d}S_{\boldsymbol{y}} = \sum_{i=1}^{N_p} \int_{-1}^{1} f(\boldsymbol{y}_i(t)) \left|\boldsymbol{y}_i'(t)\right| \mathrm{d}t \approx \sum_{i=1}^{N_p}\sum_{j=1}^{n} f(\boldsymbol{y}_i(t_j)) \left|\boldsymbol{y}_i'(t_j)\right| \lambda_j. \tag{36}$$

Here $(t_j, \lambda_j)_{j=1}^n$ are the standard Gauss-Legendre nodes and weights on $[-1, 1]$. For convenience, we now combine the double summation over i and j into a single index m that runs to $N_{\partial\Omega} = nN_p$, and let $\boldsymbol{x}_m = \boldsymbol{y}_i(t_j)$ and $w_m = \left|\boldsymbol{y}_i'(t_j)\right| \lambda_j$. This gives us a set of nodes and weights $(\boldsymbol{x}_m, w_m)_{m=1}^{N_{\partial\Omega}}$ for approximating integrals over $\partial\Omega$. Given the double layer density $\boldsymbol{\mu}$ at the nodes, we can then approximate the double layer potential (29) anywhere in Ω as

$$\tilde{\mathcal{D}}_i[\boldsymbol{\mu}](\boldsymbol{x}) = \sum_{m=1}^{N_{\partial\Omega}} D_{ij}(\boldsymbol{x}, \boldsymbol{x}_m)\mu_j(\boldsymbol{x}_m) w_m, \quad \boldsymbol{x} \in \Omega. \tag{37}$$

As long as $\partial\Omega$ and $\boldsymbol{\mu}$ are properly resolved, this quadrature will be accurate for target points $\boldsymbol{x}$ that are well-separated from $\partial\Omega$. However, straight Gauss-Legendre quadrature will not be accurate for target points close to, or on, the boundary $\partial\Omega$. This is due to the singularities in the stresslet (14). We will discuss specialized quadrature methods for such cases in section 4. For now, we write $w_m = w_m(\boldsymbol{x})$, with the understanding that these weights are only dependent on $\boldsymbol{x}$ at the finite number of boundary points $\boldsymbol{x}_m$ located close to $\boldsymbol{x}$.

We approximate the integral equation using the quadrature (37), including specialized quadrature weights where needed, and enforce it at the quadrature nodes (this is known as the Nyström method). This gives us the $2N_{\partial\Omega} \times 2N_{\partial\Omega}$ dense linear system

$$\frac{1}{2}\boldsymbol{\mu}(\boldsymbol{x}_n) + \sum_{m=1}^{N_{\partial\Omega}} \left(D(\boldsymbol{x}_n, \boldsymbol{x}_m) + \hat{\boldsymbol{n}}(\boldsymbol{x}_n)\hat{\boldsymbol{n}}(\boldsymbol{x}_m)^T \right) \boldsymbol{\mu}(\boldsymbol{x}_m) w_m(\boldsymbol{x}_n) = \boldsymbol{f}(\boldsymbol{x}_n) - \boldsymbol{u}^P(\boldsymbol{x}_n), \quad n = 1, \dots, N_{\partial\Omega}. \tag{38}$$

As this is a second-kind integral equation, it can be solved iteratively using GMRES [57] in a number of iterations that is bounded as $N_{\partial\Omega}$ grows. We can then evaluate the homogeneous solution on the domain grid using (37). If done naively, both of these steps have quadratic costs, which is infeasible for large systems. To get around this, we make use of fast hierarchical methods, described in sections 3.2.1 and 3.2.2.

3.2.1. Fast summation

Let $N_\Omega \leq N_B$ denote the number of grid points which lie inside Ω (in Figure 1 this is the number of black dots). A naive evaluation of (37) at each of these points would require $O(N_\Omega N_{\partial\Omega})$ operations. Using the fast multipole method (FMM) [33], it is possible to reduce this computational cost to $O(N_\Omega + N_{\partial\Omega})$, by taking advantage of special structure in the sum. In this section, we describe some components of an FMM based on the special functions defined in [9].

Nomenclature. In analogy with physics, we refer to a Green's function $G(\boldsymbol{x}, \boldsymbol{y})$ as a charge located at $\boldsymbol{y}$, a derivative of G as a dipole, a second derivative of G as a quadrupole, and a third derivative of G as an octopole. By the definition of the double layer potential, the quadrature formula (37) is a sum of dipoles and octopoles for the modified biharmonic equation which are located along $\partial\Omega$. These charges and higher moments are referred to as sources and the points at which we would like to evaluate the field are called targets.

Well-separated interactions. Suppose that a collection of sources is contained in a disc of radius R with center $\boldsymbol{x}$. Then, points outside of the disc of radius $3R$ about $\boldsymbol{x}$, i.e. points which are separated by a diameter from the disc, are considered well-separated from the charges in the disc. An important property of well-separated interactions is that the field induced by a set of sources is smooth from the perspective of well-separated points and can be approximated accurately using relatively few terms in a separation of variables expansion, independent of the number of source points. In practice, charges are sorted into the boxes of a quad-tree, as described below, and an analogous notion of well-separated interactions is defined; see fig. 2a for an illustration.

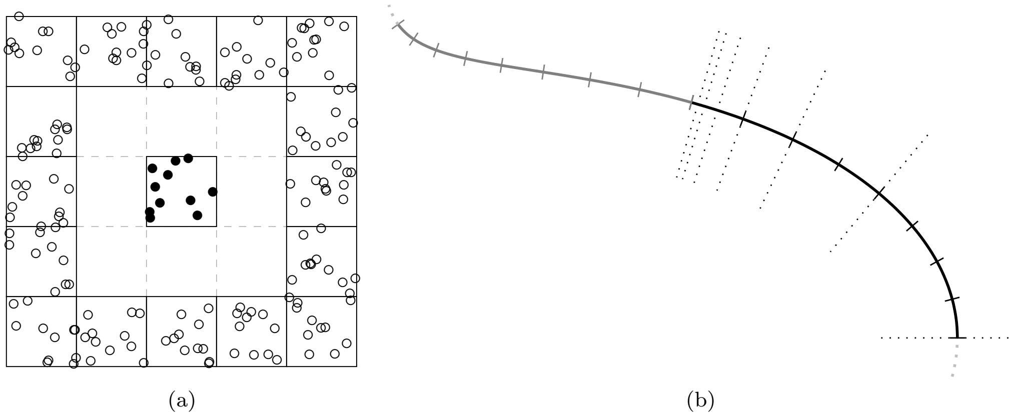


Figure 2: (a) We plot a collection of white dots which are well-separated from the black dots. (b) An illustration of the property that adjacent pieces of a one dimensional curve are of relatively low rank. The points on the gray section of the curve are well-separated from each set of points between the dotted separators on the black section of the curve.

Separation-of-variables representations. Because the field induced by the sources satisfies the modified biharmonic equation, standard separation-of-variables techniques apply. Let $I_n(\alpha r)$ and $K_n(\alpha r)$ denote the modified Bessel functions of the first and second kind, respectively. Classically, a field which satisfies the modified biharmonic equation can be represented in the exterior of the disc as a series in $\log r$, $r^{-|n|}$, and $K_n(\alpha r)$ and in the interior of a disc as a series in $r^{|n|}$ and $I_n(\alpha r)$, where r is the radial part of polar coordinates centered in the disc. Using these classical functions directly leads to catastrophic cancellation when αr is small. Following [9], we define the functions Q_n and P_n via

$$Q_n(r) = K_n(\alpha r) - \frac{2^{|n|-1}\left(|n|-1\right)!}{\alpha^{|n|} r^{|n|}} \ , \ |n| > 0 \ , \tag{39}$$

$$Q_0(r) = K_0(\alpha r) + \log(r) \ , \tag{40}$$

and

$$P_n(r) = I_n(\alpha r) - \left(\frac{\alpha r}{2}\right)^{|n|} \frac{1}{|n|!} \ . \tag{41}$$

Consider a disc centered at a point $\boldsymbol{y}$ and let (r,θ) denote the polar coordinates of a point $\boldsymbol{x}-\boldsymbol{y}$. Then, the field, ψ, induced by a collection of sources inside the disc can be represented by

$$\psi(\boldsymbol{x}) = \sum_{n=-\infty}^{\infty} \big(\alpha_n Q_n(r) + \beta_n K_n(\alpha r)\big)\, e^{in\theta} \ , \tag{42}$$

which we will refer to as a multipole expansion, and the field, ϕ, induced by a collection of sources outside the disc can be represented by

$$\phi(\boldsymbol{x}) = \sum_{n=-\infty}^{\infty} \left(a_n P_n(\alpha r) + b_n r^{|n|}\right) e^{in\theta} \ , \tag{43}$$

which we will refer to as a local expansion. These expansions can be manipulated stably. In particular, [9] provides stable formulas for shifting the centers of the expansions and converting a multipole expansion about one center into a local expansion about another. When evaluating the expansions for a collection of sources which are well separated from the targets, the standard decay rates in [32, 21] apply, so that truncating the sums in (42) and (43) to order $p_{\text{FMM}} = 40$, i.e. retaining only the terms with $|n| \leq p_{\text{FMM}}$, is sufficient to achieve 12 digits of accuracy.

Tree structure. To deal with arbitrary collections of sources and targets, as opposed to well-separated ones, the source and target points are arranged in a hierarchical quad-tree structure in the FMM. Starting from a root box containing all sources and targets, the tree nodes (boxes) are subdivided equally into four quadrants, which are its children nodes, until they have at most a prescribed total number of source and target points inside. Typically, the maximum number of points is set to the order of the expansions used to represent the well-separated interactions. Many FMM algorithms are written for arbitrary quad-trees; for simplicity, we add a level-restriction to the tree: all adjacent leaf boxes should be within one level of refinement.

The FMM gains its efficiency by organizing the interactions between sources and targets hierarchically in this tree so that most interactions are treated as well-separated. This is done using a number of standard techniques, see [32, 33] for a reference. Below, we only describe how multipole and local expansions are formed on leaf boxes, as the other techniques are a straightforward application of the translation formulas in [9].

Form multipole and form local. Consider the sources contained within a leaf box. Note that a charge can be represented perfectly as a 0-term expansion about its own center because $G(\boldsymbol{x},\boldsymbol{y}) = -Q_0(r)/(2\pi\alpha^2)$, with $r = |\boldsymbol{x}-\boldsymbol{y}|$. Because we are concerned with up to octopole moments of the Green's function, we require expressions for first, second, and third order directional derivatives of G in the multipole basis $\{Q_n, K_n\}$. We derive an expression for first order derivatives here to illustrate the idea, and provide expressions for the higher order derivatives in appendix C. As in appendix B, we make use of the standard suffix notation and set $\boldsymbol{r} = \boldsymbol{x}-\boldsymbol{y}$.

We have

$$\partial_i G(\boldsymbol{r}) = G'(r)\frac{r_i}{r} \ . \tag{44}$$

From the recurrence and derivative relations for modified Bessel functions [51, §10.29], we obtain

$$G'(r) = -\frac{Q_1(r)}{2\pi\alpha} \ . \tag{45}$$

To derive the separation of variables form of the directional derivatives, we identify the point $\boldsymbol{r}$ with $re^{i\theta}$. In the following, we switch freely between identifying vectors by their suffix and complex valued notations. Let $\boldsymbol{\lambda}$ denote a vector and λ denote its complex valued counterpart. Then,

$$\frac{r_i\lambda_i}{r} = \frac{e^{i\theta}\bar{\lambda} + e^{-i\theta}\lambda}{2} \ , \tag{46}$$

so that

$$\partial_\lambda G = -\frac{Q_1(r)}{4\pi\alpha}\left(e^{i\theta}\bar{\lambda} + e^{-i\theta}\lambda\right) \ . \tag{47}$$

Remark 2. *Note that the $e^{in\theta}$ terms always have a coefficient which is the complex conjugate of the $e^{-in\theta}$ terms. This is used by the fast multipole method to reduce the storage for expansions and to reduce the computational cost of translating expansions.*

Once the exact separation of variables representation for a charge or higher order moment of G is obtained from the formulas above, one can simply use the translation formulas for multipole expansions to shift each of the per-source expansions to the center of the leaf box. The cost of this operation depends on the number of sources, but is only performed on leaf boxes. Similarly, the per-source expansions can be translated, transformed, and added to the local expansion in any box for which the sources are well-separated.

3.2.2. Fast direct solver

Let $\boldsymbol{A}$ be the $2N_{\partial\Omega} \times 2N_{\partial\Omega}$ matrix corresponding to the discretized boundary integral equation, (38), so that

$$A_{2(i-1)+k,2(j-1)+l} = \frac{1}{2}\delta_{2(i-1)+k,2(j-1)+l} + D_{kl}(\boldsymbol{x}_i, \boldsymbol{x}_j)w_j(\boldsymbol{x}_i) \ , \tag{48}$$

with $1 \leq i, j \leq N_{\partial\Omega}$ and $1 \leq k, l \leq 2$. Naïve inversion of this matrix would require $O({N_{\partial\Omega}}^3)$ floating point operations. However, the complexity of inversion can be reduced to $O(N_{\partial\Omega})$ by taking advantage of structure within the matrix.

The ability to use a truncated multipole expansion to represent well-separated interactions to high precision, as in the previous section, implies that sub-matrices of $\boldsymbol{A}$ for which the row points are well-separated from the column points are of low numerical rank (and likewise when the column points are well-separated from the row points). Because of these low rank structures in the matrix, there is an ordering of the indices for which the matrix $\boldsymbol{A}$ is hierarchically block separable (HBS), as in [28, 48]. For such matrices, it is possible to efficiently compute a compressed representation of the matrix in $O(N_{\partial\Omega})$ operations and using $O(N_{\partial\Omega})$ storage. Then, a representation of the inverse can also be computed and applied in $O(N_{\partial\Omega})$ operations based on that compressed representation. Such methods combine matrix compression [20] with a variant of the Sherman-Woodbury-Morrison formula and take advantage of hierarchical features of the matrix, as in the FMM. See [28, 48] for more details.

A common feature of fast direct solvers is that there is a relatively large upfront cost in forming the representation of the inverse while the cost of the subsequent linear system solves for any given right hand side is orders of magnitude smaller. This is well-suited to the numerical examples in this paper, where we have a fixed boundary discretization and use a fixed time step length. We only pay the upfront cost once and use the same representation of the inverse for fast system solves at each step in the simulation.

Remark 3. *Technically, the $O(N_{\partial\Omega})$ scaling of the HBS method depends on a more subtle aspect of the low-rank structure of the matrix. Because the boundary points are contained to a one dimensional curve, even the interactions between adjacent sections of the curve are of relatively low rank. This is illustrated in fig. 2b for two adjacent sections of the boundary, one gray and one black. Observe that the points on the gray portion of curve are approximately well-separated from each set of points between the dotted separators on the black portion of the curve.*

3.3. Time stepping

The integral equation approach that we have outlined so far is based on a first order IMEX discretization of the Navier-Stokes equations (3). In order to increase the temporal order of accuracy, we use the semi-implicit spectral deferred corrections (SISDC) method [50]. Compared to the regular spectral deferred correction (SDC) method [22], which uses Gauss-Legendre nodes and either implicit or explicit Euler time marching, SISDC uses Gauss-Lobatto nodes and semi-implicit (IMEX) time marching.

Consider an autonomous initial value problem of the form

$$\phi'(t) = G(\phi(t)) = G_E(\phi(t)) + G_I(\phi(t)), \tag{49}$$

$$\phi(0) = \phi_0, \tag{50}$$

where G_I is a stiff part that needs implicit treatment, and G_E is a nonlinear term that would be impractical to treat implicitly. The method is based on dividing a time interval $[t_a, t_b]$, $t_b - t_a = \Delta t$, into p subintervals by choosing points t_m such that $t_a = t_0 < t_1 < \cdots < t_p = t_b$, with $\delta t_m = t_{m+1} - t_m$. A first approximation of $\phi(t)$ on $[t_a, t_b]$, denoted $\phi^0(t)$, is computed at the substeps $\phi^0_m = \phi^0(t_m)$ using a first order IMEX discretization,

$$\phi^0_{m+1} - \delta t_m G_I(\phi^0_{m+1}) = \phi^0_m + \delta t_m G_E(\phi^0_m), \quad m = 0, \ldots, p-1, \tag{51}$$

$$\phi^0_0 = \phi(t_a). \tag{52}$$

Then, a sequence of increasingly accurate approximations $\{\phi^1(t), \phi^2(t), \phi^3(t), \ldots\}$ are computed at the nodes $\phi^k_m = \phi^k(t_m)$ using the correction equation

$$\begin{aligned} \phi^{k+1}_{m+1} - \delta t_m G_I(\phi^{k+1}_{m+1}) = \phi^{k+1}_m + \delta t_m \left[G_E(\phi^{k+1}_m) - G_E(\phi^k_m) - G_I(\phi^k_{m+1}) \right] + I^{m+1}_m(\phi^k_0, \ldots, \phi^k_p), \\ m = 0, \ldots, p-1, \\ k = 0, 1, 2, \ldots, \end{aligned} \tag{53}$$

all initialized to $\phi^k_0 = \phi(t_a)$. The computation of $\{\phi^{k+1}_0, \ldots, \phi^{k+1}_p\}$ is in this context considered one application of the correction equation. The term $I^{m+1}_m(\phi^k_0, \ldots, \phi^k_p)$ is a quadrature approximation of the integral

$$I^{m+1}_m(\phi^k) = \int_{t_m}^{t_{m+1}} G(\phi^k(t))\, \mathrm{d}t. \tag{54}$$

The quadrature is precomputed, so that computing the integral requires only a small matrix-vector multiplication, as in

$$I^{m+1}_m(\phi^k_0, \ldots, \phi^k_p) = \sum_{j=0}^{p} q^m_j G(\phi^k_j), \quad m = 0, \ldots, p-1. \tag{55}$$

The weights q^m_j are chosen so that they integrate the interpolating polynomial through the $p+1$ points, which gives an $\mathcal{O}(\Delta t^{p+2})$ error in each quadrature. In [50] the points are chosen to be the nodes of Gauss-Lobatto quadrature. This choice ensures that the polynomial interpolation is well-conditioned for high order schemes, while conveniently including nodes at the beginning and end of the interval.

Each application of the correction equation increases the local order of accuracy by one, so that if $\phi(t_b) - \phi^0_p = \mathcal{O}(\Delta t^2)$, then the local time step error after k corrections is

$$\phi(t_b) - \phi^k_p = \mathcal{O}(\Delta t^{k+2} + \Delta t^{p+2}). \tag{56}$$

A SISDC time stepping method with global order of accuracy K therefore requires $p = K-1$ substeps and $K-1$ applications of the correction equation. Taking into account the p steps taken to compute the first approximation (51), the total number of implicit linear system solves needed to take one step of length Δt is then $K^2 - K$, resulting in the final approximation $\phi(t_b) \approx \phi^{K-1}_{K-1}$.

In our case, the equation of interest is the momentum equation of the Navier-Stokes equations (1). We split the equation into implicit and explicit parts as before, setting

$$G_E = -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}, \tag{57}$$

$$G_I = \frac{1}{Re}\Delta \boldsymbol{u} - \nabla p. \tag{58}$$

Our scheme for the modified Stokes equation relies on precomputing a solver and quadrature rules for a specific α, which in this context is a function of the substep length. To reduce the precomputation time and storage, it is therefore preferable for us to use substeps that are of equal length $\delta t = \Delta t/(K-1)$. This coincides with the points used by [50] for $K = 2$ (where $\delta t = \Delta t$) and $K = 3$ (since 3-point Gauss-Lobatto is equidistant). For moderate values of $K > 3$, equispaced substeps could still be feasible, though at some level of K the ill-conditioning of polynomial interpolation on equispaced nodes would make itself known.

Assuming a constant substep length of δt, the first approximation is computed by using our composite scheme to solve the modified Stokes equation (9),

$$(\alpha^2 - \Delta)\boldsymbol{u}_{m+1}^0 + Re\, \nabla p_{m+1}^0 = \boldsymbol{F}_{m+1}^0, \quad \alpha^2 = Re\,/\delta t, \tag{59}$$

with the forcing function from the original IMEX scheme (3),

$$\boldsymbol{F}_{m+1}^0 = \alpha^2 \boldsymbol{u}_m^0 - Re(\boldsymbol{u}_N^0 \cdot \nabla)\boldsymbol{u}_N^0. \tag{60}$$

To apply the correction equation, we solve the same PDE,

$$(\alpha^2 - \Delta)\boldsymbol{u}_{m+1}^{k+1} + Re\, \nabla p_{m+1}^{k+1} = \boldsymbol{F}_{m+1}^{k+1}, \tag{61}$$

this time with the right hand side

$$\boldsymbol{F}_{m+1}^{k+1} = \alpha^2 \boldsymbol{u}_m^{k+1} - Re\left[(\boldsymbol{u}_m^{k+1} \cdot \nabla)\boldsymbol{u}_m^{k+1} - (\boldsymbol{u}_m^k \cdot \nabla)\boldsymbol{u}_m^k\right] - \Delta \boldsymbol{u}_{m+1}^k + Re\, \nabla p_{m+1}^k + \frac{Re}{\delta t} I_m^{m+1}, \tag{62}$$

and the quadrature term

$$\frac{Re}{\delta t} I_m^{m+1} = \frac{1}{\delta t} \sum_{j=0}^{p} q_j^m \left[\Delta \boldsymbol{u}_j^k - Re\, \nabla p_j^k - Re(\boldsymbol{u}_j^k \cdot \nabla)\boldsymbol{u}_j^k\right]. \tag{63}$$

For the first order IMEX scheme, we need to compute the solution $\boldsymbol{u}$ and its gradient $\nabla \boldsymbol{u}$ on the grid at every substep, and we also need them at the initial time $t = 0$ to start the simulation. In addition to that, the SISDC scheme also requires the quantity $\Delta \boldsymbol{u} - Re\, \nabla p$ at every substep, and at the initial time. At a substep (m, k) we can compute this quantity without actually having to compute $\Delta \boldsymbol{u}_m^k$ and ∇p_m^k separately, simply by recovering it from the equation solved to compute the solution $\boldsymbol{u}_m^k$ at that substep,

$$\Delta \boldsymbol{u}_m^k - Re\, \nabla p_m^k = \alpha^2 \boldsymbol{u}_m^k - \boldsymbol{F}_m^k. \tag{64}$$

For the initial time, this quantity is not available. Note that, when added to the right hand side, any irrotational term (which is the gradient of a scalar function) has no effect on the computed velocity field. Therefore, we only require the value of $\Delta \boldsymbol{u}$ be supplied at the initial time and use $\Delta \boldsymbol{u}_0^0$ instead of $\Delta \boldsymbol{u}_0^0 - Re\, \nabla p_0^0$ in correction equations for the first interval.

Remark 4. *The fact that the irrotational part of the right hand side has no effect on the velocity is clear in two dimensions when analyzing the stream function formulation of the modified Stokes equations,* (9)*. Let* $\boldsymbol{u} = \nabla^\perp \Psi$ *and write the Helmholtz decomposition of the right hand side* $\boldsymbol{F}$ *as* $\boldsymbol{F} = \nabla \Phi + \nabla^\perp \Xi$ *for some scalar functions* Ψ*,* Φ*, and* Ξ*. Substituting these expressions into* (9) *and taking the dot product of both sides with* $\nabla^\perp$*, we obtain*

$$(\alpha^2 - \Delta)\Delta\Psi = \Delta\Xi\ ,$$

which depends only on the solenoidal part of $\boldsymbol{F}$*.*

4. Quadrature details

As mentioned in section 2.2, composite Gauss-Legendre quadrature over $\partial\Omega$ is only accurate for target points that are well-separated from $\partial\Omega$. More precisely, the contribution to the double layer potential from a particular panel Γ using the Gauss-Legendre quadrature, which we will refer to as the *direct quadrature*, is only accurate if the target point $\boldsymbol{x}$ is sufficiently far away from that panel. Our goal, then, is to have a quadrature that can accurate evaluate

$$\int_\Gamma \mu_i(\boldsymbol{y}) T_{ijk}(\boldsymbol{x},\boldsymbol{y}) \hat{n}_k(\boldsymbol{y}) \,\mathrm{d}S_{\boldsymbol{y}}, \tag{65}$$

for any panel $\Gamma \subset \partial\Omega$ and target point $\boldsymbol{x}$. For this, we will distinguish between two different cases: the singular case, when $\boldsymbol{x} \in \partial\Omega$, and the nearly singular case, when $\boldsymbol{x} \in \Omega$ is close to $\partial\Omega$.

Our method of choice for the singular and nearly singular quadrature will be the kernel-split, complex interpolatory quadrature scheme of Helsing et al. [39, 36, 37]. This scheme is both accurate and efficient. In addition, at each target point it only requires a finite (and small) number of operations to correct the direct quadrature. As the direct quadrature sum is what is returned by the FMM, the kernel-split quadrature scheme is a natural and efficient complement to the FMM; it is *FMM-compatible*, according to the definition of [35].

4.1. Kernel split

The core problem that kernel-split quadrature addresses is that the stresslet contains several singularities as $\boldsymbol{x} \to \boldsymbol{y}$. In order to apply a kernel-split quadrature, we must first rewrite the kernel in a way that explicitly exposes these singularities. The closed-form expression for the stresslet, derived in appendix B, is

$$T_{ijk}(\boldsymbol{x},\boldsymbol{y}) = \alpha^2 \mathcal{T}_1(\alpha\|\boldsymbol{r}\|)\left(\delta_{jk} r_i + \delta_{ik} r_j + \delta_{ij} r_k\right) + \alpha^4 \mathcal{T}_2(\alpha\|\boldsymbol{r}\|) r_i r_j r_k + \alpha^2 \mathcal{T}_3(\alpha\|\boldsymbol{r}\|)\delta_{ik} r_j, \tag{66}$$

where $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}$, and the stresslet functions $\mathcal{T}_1$–$\mathcal{T}_3$ can be expressed in terms of modified Bessel functions of the second kind,

$$\mathcal{T}_1(z) = -\frac{2z^2 K_0(z) + \left(z^2+4\right) z K_1(z) - 4}{2\pi z^4}, \tag{67}$$

$$\mathcal{T}_2(z) = \frac{4z^2 K_0(z) + \left(z^2+8\right) z K_1(z) - 8}{\pi z^6}, \tag{68}$$

$$\mathcal{T}_3(z) = \frac{z K_1(z) - 1}{2\pi z^2}. \tag{69}$$

It is clear that any singular behavior in the stresslet must come from the functions $\mathcal{T}_i$. Using a standard result [51, §10.31], we can decompose, or split, K_0 and K_1 into explicit singularities as

$$K_0(z) = K_0^S(z) - I_0(z)\log z, \tag{70}$$

$$K_1(z) = K_1^S(z) + I_1(z)\log z + \frac{1}{z}. \tag{71}$$

Here $K_0^S(z)$ and $K_1^S(z)$ are simply the remainders after removing the singularities; both of these terms, along with I_0 and I_1 (modified Bessel functions of the first kind), are smooth functions of z. Using this, we can now split the stresslet functions $\mathcal{T}_i$ into smooth functions multiplying explicit singularities,

$$\mathcal{T}_1(z) = \mathcal{T}_1^S(z) + \mathcal{T}_1^L(z)\log z, \tag{72}$$

$$\mathcal{T}_2(z) = \mathcal{T}_2^S(z) + \mathcal{T}_2^L(z)\log z + \frac{1}{8\pi z^2} - \frac{1}{\pi z^4}, \tag{73}$$

$$\mathcal{T}_3(z) = \mathcal{T}_3^S(z) + \mathcal{T}_3^L(z)\log z, \tag{74}$$

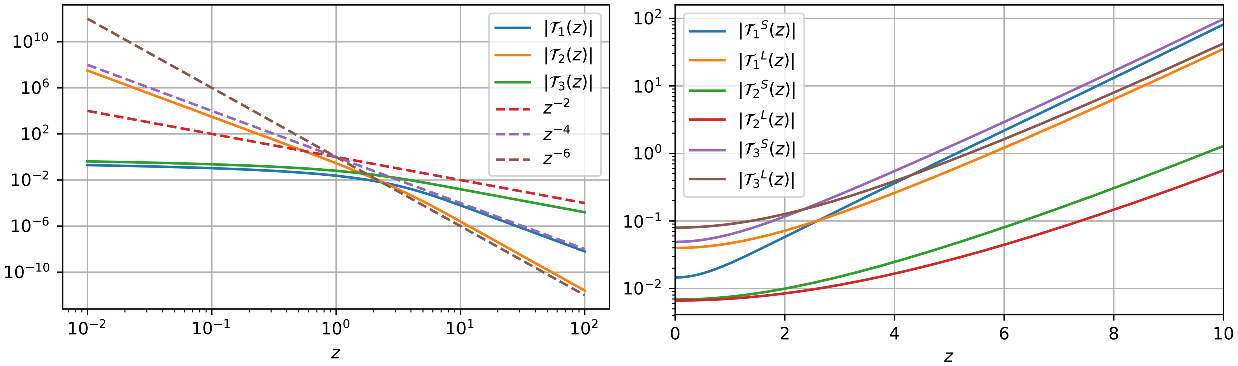


Figure 3: Left: The stresslet functions $\mathcal{T}_1$–$\mathcal{T}_3$, as defined in (67)–(69). Right: The splits $\mathcal{T}_1^S$–$\mathcal{T}_3^L$, as defined in (72)–(74). Note that the magnitudes of the stresslet functions decay algebraically as $z \to \infty$, even though the magnitudes of their decompositions grow exponentially.

where

$$\mathcal{T}_1^S(z) = -\frac{2zK_0^S(z) + \left(z^2+4\right) K_1^S(z) + z}{2\pi z^3}, \qquad \mathcal{T}_1^L(z) = \frac{2zI_0(z) - \left(z^2+4\right) I_1(z)}{2\pi z^3}, \tag{75}$$

$$\mathcal{T}_2^S(z) = \frac{32zK_0^S(z) + 8\left(z^2+8\right) K_1^S(z) - z\left(z^2-16\right)}{8\pi z^5} \qquad \mathcal{T}_2^L(z) = \frac{\left(z^2+8\right) I_1(z) - 4zI_0(z)}{\pi z^5}, \tag{76}$$

$$\mathcal{T}_3^S(z) = \frac{K_1^S(z)}{2\pi z} \qquad \mathcal{T}_3^L(z) = \frac{I_1(z)}{2\pi z}. \tag{77}$$

All of the above expressions are prone to cancellation for small values of $z = \alpha\|\boldsymbol{r}\|$, and must then be evaluated using power series expansions, discussed further in appendix D.

Using the above definitions, we now split the stresslet as

$$T_{ijk}(\boldsymbol{r}) = \Big(T_{ijk}^S(\boldsymbol{r}) + T_{ijk}^L(\boldsymbol{r})\log(\alpha)\Big) + T_{ijk}^L(\boldsymbol{r})\log\|\boldsymbol{r}\| + T_{ij}^C(\boldsymbol{r})\frac{r_k}{\|\boldsymbol{r}\|^2} + T^Q\frac{r_ir_jr_k}{\|\boldsymbol{r}\|^4}, \tag{78}$$

where

$$\begin{aligned}
T_{ijk}^S(\boldsymbol{r}) &= \alpha^2\mathcal{T}_1^S(\alpha\|\boldsymbol{r}\|)\left(\delta_{jk}r_i + \delta_{ik}r_j + \delta_{ij}r_k\right) + \alpha^4\mathcal{T}_2^S(\alpha\|\boldsymbol{r}\|)r_ir_jr_k + \alpha^2\mathcal{T}_3^S(\alpha\|\boldsymbol{r}\|)\delta_{ik}r_j, && (79)\\
T_{ijk}^L(\boldsymbol{r}) &= \alpha^2\mathcal{T}_1^L(\alpha\|\boldsymbol{r}\|)\left(\delta_{jk}r_i + \delta_{ik}r_j + \delta_{ij}r_k\right) + \alpha^4\mathcal{T}_2^L(\alpha\|\boldsymbol{r}\|)r_ir_jr_k + \alpha^2\mathcal{T}_3^L(\alpha\|\boldsymbol{r}\|)\delta_{ik}r_j, && (80)\\
T_{ij}^C(\boldsymbol{r}) &= \alpha^2 r_ir_j/8\pi, && (81)\\
T^Q &= -1/\pi, && (82)
\end{aligned}$$

Using these results, together with the relation $T(\boldsymbol{x},\boldsymbol{y}) = -T(\boldsymbol{y},\boldsymbol{x})$, we write the double layer potential contribution (29) from a panel Γ as

$$\int_\Gamma \mu_i(\boldsymbol{y})\left(T_{ijk}^S(\boldsymbol{r}) + T_{ijk}^L(\boldsymbol{r})\log\|\boldsymbol{r}\| + T_{ij}^C(\boldsymbol{r})\frac{r_k}{\|\boldsymbol{r}\|^2} + T^Q\frac{r_ir_jr_k}{\|\boldsymbol{r}\|^4}\right)\hat{n}_k(\boldsymbol{y})\,\mathrm{d}S_{\boldsymbol{y}}, \quad \boldsymbol{r} = \boldsymbol{y} - \boldsymbol{x}. \tag{83}$$

This form, where the integrand is a smooth function plus a sum of smooth functions multiplied by explicit singularities, is what allows us to design efficient quadrature.

4.2. On-boundary evaluation of the double layer potential

When solving the discretized integral equation (38), we need to evaluate the double layer potential for target points $\boldsymbol{x}$ on the boundary $\partial\Omega$. If $\boldsymbol{x}$ is close to Γ but distant in arclength, then $\boldsymbol{x}$ is treated as a nearly singular point, which we will return to in section 4.3. This can happen, for example, if $\partial\Omega$ is multiply-connected or it has a narrow channel. If on the other hand $\boldsymbol{x}$ is close to Γ both in distance and arclength, which in practice means that $\boldsymbol{x}$ is on Γ or on a neighboring panel, then we must consider what happens with the singularities in the double layer potential (83) in the limit $\boldsymbol{x} \to \boldsymbol{y}$ on $\partial\Omega$: the $\log\|\boldsymbol{r}\|$-term is singular in the limit, but that singularity is cancelled by $T^L \to 0$. The $\boldsymbol{r}\cdot\hat{\boldsymbol{n}}/\|\boldsymbol{r}\|^2$-term has a finite limit, which is then removed by $T^C \to 0$. The smooth term T^S also disappears, so all that remains is the smooth limit of the fourth term,

$$\lim_{\substack{\boldsymbol{x}\to\boldsymbol{y}\\ \boldsymbol{x},\boldsymbol{y}\in\partial\Omega}} T_{ijk}(\boldsymbol{y}-\boldsymbol{x})\hat{n}_k(\boldsymbol{y}) = \frac{\hat{t}_i(\boldsymbol{y})\hat{t}_j(\boldsymbol{y})\kappa(\boldsymbol{y})}{2\pi}, \tag{84}$$

where $\hat{\boldsymbol{t}} = (\hat{n}_2, -\hat{n}_1)$ is the unit tangent vector and κ is the local curvature. However, even though the log-singularity is cancelled out by its multiplying factor, we must still apply kernel-split quadrature to it, in order to evaluate it to high accuracy. This is done in precisely the same way as for near-boundary evaluation in section 4.3. The remaining three terms in (83) can be evaluated directly using Gauss-Legendre quadrature, as they are smooth on $\partial\Omega$.

4.3. Near-boundary evaluation of the double layer potential

For a target point $\boldsymbol{x} \in \Omega$ close to Γ, the kernel of (83) has singularities close to the interval of integration, which are often referred to as near-singularities in the literature. This makes the integrand a bad candidate for polynomial approximation, which is why direct Gauss-Legendre quadrature fails. The kernel-split quadrature, by contrast, computes weights for each term of (83) separately, and works well as long as the function multiplying the singularity is smooth. The scheme works by first rewriting the singularities in complex form, resulting in linear combinations of the integrals

$$\int_\Gamma f(\tau)\log(\tau - z)\,\mathrm{d}\tau \quad \text{and} \quad \int_\Gamma f(\tau)\frac{\mathrm{d}\tau}{(\tau - z)^m},\ m = 1, 2, \dots. \tag{85}$$

Here $z = x_1 + ix_2$ and $\tau = y_1 + iy_2$ are complex representations of the target point and the panel, and f is assumed to be a smooth, real-valued function. Then, for a panel with n nodes, f is approximated using a monomial expansion,

$$f(\tau) \approx \sum_{k=1}^{n} c_k \tau^{k-1}, \tag{86}$$

such that, e.g.,

$$\int_\Gamma f(\tau)\frac{\mathrm{d}\tau}{(\tau - z)^m} \approx \sum_{k=1}^{n} c_k p_k, \quad \text{where} \quad p_k = \int_\Gamma \frac{\tau^{k-1}\,\mathrm{d}\tau}{(\tau - z)^m}. \tag{87}$$

The integrals p_k can be evaluated using recursion formulas. By solving a transposed $n \times n$ Vandermonde system, quadrature weights w_j can be computed [39], such that the integral of a function f known at the nodes τ_j can be approximated as

$$\int_\Gamma f(\tau)\frac{\mathrm{d}\tau}{(\tau - z)^m} \approx \sum_{j=1}^{n} w_j f(\tau_j). \tag{88}$$

This quadrature is nth order accurate, as it is based on interpolation of f at the n quadrature nodes. Solution of the complex $n \times n$ Vandermonde system can be accelerated using the Björck-Pereyra algorithm [16].

In our layer potential (83), the first term is smooth and can be evaluated using the direct Gauss-Legendre quadrature. The remaining three terms must be converted into the forms (85) before we can evaluate them. For this, we use the following notation:

$$\begin{aligned} z &= x_1 + ix_2, & \tau(t) &= y_1(t) + iy_2(t), \\ \nu_\tau &= i\tau'(t)/\left|\tau'(t)\right|, & \omega &= \mu_1 + i\mu_2. \end{aligned}$$

Here $\boldsymbol{y}(t)$ is the parametrization of Γ, as defined in 3.2. Identifying $\mathbb{R}^2$ with $\mathbb{C}$ and using $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{x} = \tau - z$, the remaining three terms can now be cast into complex forms using the relations

$$\int_\Gamma f(\boldsymbol{x}, \boldsymbol{y}) \log\|\boldsymbol{r}\| \,\mathrm{d}S = -\operatorname{Im} \int_\Gamma f(z, \tau) \bar{\nu}_\tau \log(\tau - z) \,\mathrm{d}\tau, \tag{89}$$

$$\int_\Gamma f(\boldsymbol{x}, \boldsymbol{y}) \frac{\boldsymbol{r} \cdot \hat{\boldsymbol{n}}}{\|r\|^2} \,\mathrm{d}S = -\operatorname{Im} \int_\Gamma f(z, \tau) \frac{\mathrm{d}\tau}{\tau - z}, \tag{90}$$

$$\int_\Gamma \frac{(\boldsymbol{\mu} \cdot \boldsymbol{r})\boldsymbol{r}(\boldsymbol{r} \cdot \hat{\boldsymbol{n}})}{\|\boldsymbol{r}\|^2} \,\mathrm{d}S = -\frac{1}{2} \int_\Gamma \omega \operatorname{Im}[(\bar{\tau} - \bar{z}) \,\mathrm{d}\tau] - \frac{1}{4i} \int_\Gamma \bar{\omega}(\tau - z) \,\mathrm{d}\tau + \frac{1}{4i} \overline{\int_\Gamma \frac{\omega(\bar{\tau} - \bar{z})^2}{(\tau - z)} \,\mathrm{d}\tau}, \tag{91}$$

$$\int_\Gamma \frac{(\boldsymbol{\mu} \cdot \boldsymbol{r})\boldsymbol{r}(\boldsymbol{r} \cdot \hat{\boldsymbol{n}})}{\|\boldsymbol{r}\|^4} \,\mathrm{d}S = -\frac{1}{2} \int_\Gamma \omega \operatorname{Im}\left[\frac{\mathrm{d}\tau}{(\tau - z}\right] + \frac{1}{4i} \overline{\int_\Gamma \frac{\omega \bar{\nu}^2 \,\mathrm{d}\tau}{\tau - z}} + \frac{1}{4i} \overline{\int_\Gamma \frac{\omega(\bar{\tau} - \bar{z}) \,\mathrm{d}\tau}{(\tau - z)^2}}. \tag{92}$$

In order to use the relation (89) to compute the second term of (83), we set $f = \mu_i T^L_{ijk} \hat{n}_k$ for $j = 1, 2$. The third term of (83) can be computed either using (90) with $f = \alpha^2 \mu_i r_i r_j / 8\pi$, $j = 1, 2$, or (91). We use (90), since it corresponds to the Laplace double layer potential, for which there is code readily available in [37]. The fourth term of (83) corresponds directly to (92), and is actually the stresslet of regular Stokes flow, for which quadrature was developed in [52].

By applying the above relations, and computing quadrature weights for the kernels $\log(\tau - z)$, $(\tau - z)^{-1}$, and $(\tau - z)^{-2}$, we can compute quadrature weights that accurately evaluate all the terms of (83) for $\boldsymbol{x}$ arbitrarily close to Γ.

4.4. Near-boundary evaluation of the gradient of the double layer potential

In order to get accurate values of the gradient of the double layer potential, we need to apply the above procedure to a kernel-split form of the integral

$$\frac{\partial}{\partial x_l} \int_\Gamma \mu_i(\boldsymbol{y}) \left(T^S_{ijk}(\boldsymbol{r}) + T^L_{ijk}(\boldsymbol{r}) \log\|\boldsymbol{r}\| + \frac{\alpha^2}{8\pi} \frac{r_i r_j r_k}{\|\boldsymbol{r}\|^2} - \frac{1}{\pi} \frac{r_i r_j r_k}{\|\boldsymbol{r}\|^4} \right) \hat{n}_k(\boldsymbol{y}) \,\mathrm{d}S_{\boldsymbol{y}}, \quad \boldsymbol{r} = \boldsymbol{y} - \boldsymbol{x}. \tag{93}$$

Direct differentiation of the first term of (93) gives

$$\begin{aligned} -\frac{\partial}{\partial x_\ell} T^S_{ijk}(\boldsymbol{r}) = & \frac{\alpha r_\ell}{\|\boldsymbol{r}\|} \left(\alpha^2 \mathcal{T}^{S,\prime}_1(\alpha\|\boldsymbol{r}\|) \left(\delta_{jk} r_i + \delta_{ik} r_j + \delta_{ij} r_k\right) + \alpha^4 \mathcal{T}^{S,\prime}_2(\alpha\|\boldsymbol{r}\|) r_i r_j r_k + \alpha^2 \mathcal{T}^{S,\prime}_3(\alpha\|\boldsymbol{r}\|) \delta_{ik} r_j \right) \\ & + \alpha^2 \mathcal{T}^S_1(\alpha\|\boldsymbol{r}\|) \left(\delta_{jk}\delta_{i\ell} + \delta_{ik}\delta_{j\ell} + \delta_{ij}\delta_{k\ell}\right) \\ & + \alpha^4 \mathcal{T}^S_2(\alpha\|\boldsymbol{r}\|) \left(r_j r_k \delta_{i\ell} + r_i r_k \delta_{j\ell} + r_i r_j \delta_{k\ell}\right) + \alpha^2 \mathcal{T}^S_3(\alpha\|\boldsymbol{r}\|) \delta_{ik}\delta_{j\ell}. \end{aligned} \tag{94}$$

This term is smooth, in spite of the leading $\|\boldsymbol{r}\|^{-1}$, since the leading order term of $\mathcal{T}^{S,\prime}_n(\alpha\|\boldsymbol{r}\|)$ is $\mathcal{O}(\|\boldsymbol{r}\|)$ (see appendix D). It can therefore be evaluated using direct Gauss-Legendre quadrature.

Direct differentiation of the second term of (93) gives

$$\frac{\partial}{\partial x_\ell} T^L_{ijk}(\boldsymbol{r}) \log\|\boldsymbol{r}\| = \frac{\partial T^L_{ijk}(\boldsymbol{r})}{\partial x_\ell} \log\|\boldsymbol{r}\| - T^L_{ijk}(\boldsymbol{r}) \frac{r_l}{\|\boldsymbol{r}\|^2}. \tag{95}$$

The derivative of T^L is computed analogously to (94), and the first term of (95) is then integrated using the same rule as for the log term of (83). The second term of (95) is integrated using the complex interpolatory quadrature for $(\tau - z)^{-1}$, after the rewrite

$$\int_\Gamma f(\boldsymbol{x}, \boldsymbol{y}) \frac{\boldsymbol{y} - \boldsymbol{x}}{\|\boldsymbol{y} - \boldsymbol{x}\|^2} \,\mathrm{d}S_{\boldsymbol{y}} = \overline{\int_\Gamma \frac{i f(z, \tau)}{\nu_\tau} \frac{\mathrm{d}\tau}{\tau - z}}, \tag{96}$$

using $f = \mu_i T^L_{ijk} \hat{n}_k$ for $j = 1, 2$.

In order to rewrite the third and fourth terms of (93), we will differentiate the complex forms (91) and (92). In general, if the velocity field is represented by a complex field ϕ,

$$u_1 + iu_2 = \phi(z, \bar{z}), \tag{97}$$

then we can get the partial derivatives as

$$\frac{\partial}{\partial x_1}(u_1 + iu_2) = \frac{\partial \phi}{\partial z} + \frac{\partial \phi}{\partial \bar{z}}, \qquad \frac{\partial}{\partial x_2}(u_1 + iu_2) = i\left(\frac{\partial \phi}{\partial z} - \frac{\partial \phi}{\partial \bar{z}}\right), \tag{98}$$

since

$$\frac{\partial z}{\partial x_1} = \frac{\partial \bar{z}}{\partial x_1} = 1, \qquad \frac{\partial z}{\partial x_2} = i, \qquad \frac{\partial \bar{z}}{\partial x_2} = -i. \tag{99}$$

Beginning with term three of (93), we identify

$$\phi = 4i \int_\Gamma \frac{\mu_i r_i r_j r_k \hat{n}_k}{\|\boldsymbol{r}\|^2} \,\mathrm{d}S. \tag{100}$$

In complex form, using (91), we have

$$\phi(z, \bar{z}) = -\int_\Gamma \omega(\bar{\tau} - \bar{z}) \,\mathrm{d}\tau + \int_\Gamma \omega(\tau - z) \,\mathrm{d}\bar{\tau} - \int_\Gamma \bar{\omega}(\tau - z) \,\mathrm{d}\tau + \overline{\int_\Gamma \frac{\omega(\bar{\tau} - \bar{z})^2}{(\tau - z)} \,\mathrm{d}\tau}, \tag{101}$$

with partial derivatives that are straightforward to evaluate,

$$\frac{\partial \phi}{\partial z} = 2i \,\mathrm{Im} \int_\Gamma \bar{\omega} \,\mathrm{d}\tau - 2\overline{\int_\Gamma \frac{\omega(\bar{\tau} - \bar{z})}{(\tau - z)} \,\mathrm{d}\tau}, \tag{102}$$

$$\frac{\partial \phi}{\partial \bar{z}} = \int_\Gamma \omega \,\mathrm{d}\tau + \overline{\int_\Gamma \frac{\omega(\bar{\tau} - \bar{z})^2}{(\tau - z)^2} \,\mathrm{d}\tau}. \tag{103}$$

Lastly, for the fourth term of (93), we identify

$$\phi = 4i \int_\Gamma \frac{\mu_i r_i r_j r_k \hat{n}_k}{\|\boldsymbol{r}\|^4} \,\mathrm{d}S. \tag{104}$$

Written in complex form, using (92), we have

$$4\pi i \phi(z, \bar{z}) = -\int_\Gamma \omega \frac{\mathrm{d}\tau}{\tau - z} + \int_\Gamma \omega \frac{\mathrm{d}\bar{\tau}}{\bar{\tau} - \bar{z}} + \overline{\int_\Gamma \frac{\omega \bar{\nu}^2 \,\mathrm{d}\tau}{\tau - z}} + \overline{\int_\Gamma \frac{\omega(\bar{\tau} - \bar{z}) \,\mathrm{d}\tau}{(\tau - z)^2}} \tag{105}$$

$$= -\int_\Gamma \frac{\omega \,\mathrm{d}\tau}{\tau - z} + \overline{\int_\Gamma \frac{(\bar{\omega} + \omega \bar{\nu}^2) \,\mathrm{d}\tau}{\tau - z}} + \overline{\int_\Gamma \frac{\omega(\bar{\tau} - \bar{z}) \,\mathrm{d}\tau}{(\tau - z)^2}}. \tag{106}$$

and

$$\frac{\partial \phi}{\partial z} = -\int_\Gamma \frac{\omega \,\mathrm{d}\tau}{(\tau - z)^2} - \overline{\int_\Gamma \frac{\omega \,\mathrm{d}\tau}{(\tau - z)^2}} = -2 \,\mathrm{Re} \int_\Gamma \frac{\omega \,\mathrm{d}\tau}{(\tau - z)^2}, \tag{107}$$

$$\frac{\partial \phi}{\partial \bar{z}} = \overline{\int_\Gamma \frac{(\bar{\omega} + \omega \bar{\nu}^2) \,\mathrm{d}\tau}{(\tau - z)^2}} + 2\overline{\int_\Gamma \frac{\omega(\bar{\tau} - \bar{z}) \,\mathrm{d}\tau}{(\tau - z)^3}}. \tag{108}$$

The last term has a singularity of type $(\tau - z)^{-3}$, which is one order higher than what we have had to evaluate so far. While the complex interpolatory quadrature scheme can in principle by used for any order singularity, the accuracy can be expected to decrease with higher order singularities. We therefore reduce the order of the singularity by one through integration by parts, such that

$$\frac{\partial \phi}{\partial \bar{z}} = -\left[\frac{\bar{\omega}(\tau - z)}{(\bar{\tau} - \bar{z})^2} \right]_{\tau_a}^{\tau_b} + \overline{\int_\Gamma \frac{\frac{\mathrm{d}\omega}{\mathrm{d}\tau}(\bar{\tau} - \bar{z}) + \omega \frac{\mathrm{d}\bar{\tau}}{\mathrm{d}\tau}}{(\tau - z)^2} \, \mathrm{d}\tau}. \tag{109}$$

To evaluate this, we make use of the relation $\frac{\mathrm{d}\bar{\tau}}{\mathrm{d}\tau} = -\bar{\nu}^2$. The derivative $\frac{\mathrm{d}\omega}{\mathrm{d}\tau}$ is evaluated using polynomial interpolation and differentiation on the panel.

4.5. Implementation details

The quadrature described above is capable of evaluating the double layer potential to high accuracy everywhere, both on and off the boundary. However, the implementation must be carried out carefully in order to obtain maximum accuracy and efficiency. We outline the full scheme in this subsection.

4.5.1. Determining where to apply kernel-split quadrature

A key element in implementing the kernel-split quadrature is determining where to apply it, which translates into determining which points are too close for the direct Gauss-Legendre quadrature, both on and off the boundary. This is important, since setting the limit too close to the source panel results in quadrature errors from the singularities, while setting the limit too far away can cause numerical errors in the kernel-split quadrature (although there is a forgiving middle ground). To make this choice, we use the quadrature error results in [5]. For a given source panel Γ, we consider the mapping $\gamma : \mathbb{R} \to \mathbb{C}$ which takes the standard interval $[-1, 1]$ to Γ. For integrals of the type (85), the quadrature error at a target point z close to Γ is dependent on the location of the preimage of z under a complexification of γ. For $t^* \in \mathbb{C}$ such that $z = \gamma(t^*)$, the quadrature error at z is approximately proportional to $\rho(t^*)^{-2n}$, where $\rho(t) = \left| t \pm \sqrt{t^2 - 1} \right|$, with the sign defined such that $\rho > 1$, is the Bernstein radius. It is possible to derive accurate quadrature error estimates for the singularities of our kernel (83), by following the steps of the error analyses in [4, 5]. However, it is for our purposes sufficient to empirically determine a limiting radius R_n, and then apply kernel split quadrature for points z such that $\rho(\gamma^{-1}(z)) < R_n$, in order to guarantee a certain quadrature precision. By solving a reference problem, such as the one in section 5.2, it is straightforward to determine R_n, for a given n, by way of manual tuning. For full numerical accuracy when evaluating our kernel, we have determined it sufficient to use $R_{16} = 3.5$ and $R_{32} = \sqrt{R_{16}}$. Note that finding $t^* = \gamma^{-1}(z)$ is a cheap operation, using the Newton-based method introduced in [5] and further discussed in [3].

4.5.2. On-boundary evaluation

For the on-boundary evaluation outlined in section 4.2, we apply the kernel-split quadrature for the $\log(\tau - z)$ singularity, using the $n = 16$ discretization points. For full numerical accuracy at $n = 16$, we have determined that this is necessary for target points z that lie on the source panel Γ_i, and for target points z on neighboring panels $\Gamma_{i\pm1}$ such that $\gamma_i^{-1}(z) \in [-1.4, 1.4]$. Note that we do not need a Newton solve to find $t^* = \gamma_i^{-1}(z)$, since we from the discretization know t such that $\gamma_{i\pm1}(t) = z$ (it is one of the Gauss-Legendre nodes on $[-1, 1]$), and γ_i and $\gamma_{i\pm1}$ describe the same function, differing only by a known linear scaling of the argument, $\gamma_i(t) = \gamma_{i\pm1}(at + b)$.

4.5.3. Near-boundary evaluation

For the near-boundary evaluation of sections 4.3 and 4.4, we determine which point-panel pairs to evaluate using kernel-split quadrature in a series of steps. For a given target point z, we first perform a coarse filtering by determining which panels are within $1.2h$ of z, where h is the panel length. This distance is experimentally determined as an upper bound on the points-panel distance where kernel-split quadrature may be required, for full accuracy at $n = 16$. This coarse filtering is implemented by first sorting the boundary points into bins of size h_{max} (the maximum panel length on $\partial\Omega$), and then computing the distances to all boundary points in bins adjacent to that of the target point. For a panel Γ satisfying this first criterion, we next perform a fine filtering by determining the Bernstein radius ρ of the preimage of z, under that panel's parametrization.

If $\rho \geq R_{16} = 3.5$, then the underlying 16-point Gauss-Legendre quadrature is accurate at z, and no special treatment is necessary. Otherwise, we proceed by first interpolating all the quantities on Γ to the nodes of a 32-point Gauss-Legendre quadrature, using barycentric Lagrange interpolation [13] (we refer to this as *upsampling* Γ). We then evaluate the quadrature from these 32 points using either the Gauss-Legendre weights, or, if $\rho < R_{32} = \sqrt{3.5}$, weights computed using the kernel-split scheme. This near-boundary evaluation can be summarized as:

- If $\rho \geq 3.5$: Use the 16-point Gauss-Legendre quadrature on the underlying panel.
- If $\sqrt{3.5} \leq \rho < 3.5$: Use the 32-point Gauss-Legendre quadrature on the upsampled panel.
- If $\rho < \sqrt{3.5}$: Use kernel-split quadrature on the upsampled panel.

The upsampling to 32-points panels in the near evaluation is useful for two reasons: It increases the accuracy of the kernel-split quadrature [3], and it increases speed by reducing the number of points where kernel-split quadrature is necessary.

To make near-boundary evaluation efficient when time stepping, we precompute and store a set of correction weights for each point-panel interaction pair that is not accurately evaluated using direct 16-point Gauss-Legendre quadrature. This allows us to first compute the direct quadrature everywhere in Ω using the FMM of section 3.2.1, and then use the correction weights to subtract off the direct quadrature and add the corrected quadrature for the points that are close to the boundary. The cost of this application is very small in relation to the cost of the FMM. The required storage is also small, since the precomputed weights for evaluating both layer potential and gradient only requires storage of $6 \times 2 \times 16$ values for each point-panel pair. In addition, most near evaluation points interact with only one or two panels, and we only need to do this for a small band of points close to $\partial\Omega$.

For the above near-boundary evaluation scheme to be robust for any target point in Ω, two special cases must be considered:

- The first special case occurs when a target point is very close to one of the discretization nodes on $\partial\Omega$. Then, first adding the direct contribution through the FMM, before subtracting it off again in the correction weights, introduces a cancellation error. This error is $\mathcal{O}(r^{-1})$ in $\boldsymbol{u}$ and $\mathcal{O}(r^{-2})$ in $\nabla\boldsymbol{u}$, where r is the distance between the target point and the nearest boundary point. To avoid this, we modify the FMM to ignore direct interactions when $r < 0.05h$, where h is the length of the panel to which the source point belongs.
- The second special case occurs when a target point is very close to an edge between two panels. In this case, cancellation errors occur in the computation of the exact integrals p_k (87) used in the complex interpolatory quadrature. To avoid this, we use a variation of the panel-merging strategy suggested in [39]. The two neighboring 16-points panels are merged, using interpolation of boundary data, into a temporary 32-point panel, from which the quadrature is evaluated as described above. This is done when the distance r between target point and panel edge satisfies the empirically determined criterion $r < 0.05 \min(h_1, h_2)$, where h_1 and h_2 are the lengths of the two nearby panels.

4.5.4. Dependence on the parameter α

Recall that the modified Stokes equation (5) depends on the parameter $\alpha = \sqrt{Re/\delta t}$, which we can expect to vary by orders of magnitude for different values of Re and δt. For our method to be robust, our quadrature scheme needs to be able to accurate integrate the stresslet T for a wide range of α. Considering the explicit form (66), having large or small values of α does not appear to be a major difficulty, as it mainly rescales the action of the kernel. However, even though the stresslet functions $\mathcal{T}_1$–$\mathcal{T}_3$ decay algebraically with $z = \alpha\|\boldsymbol{x} - \boldsymbol{y}\|$, the functions $\mathcal{T}_1^S$–$\mathcal{T}_3^L$ into which they are decomposed in (72)–(74) grow exponentially with z (see fig. 3). The root cause of this can be found in the asymptotic result $I_0(z) \sim I_1(z) \sim e^z/\sqrt{2\pi z}$ when $z \to \infty$ [51, §10.30]. From a numerical point of view, this is troublesome. Recall that the kernel-split

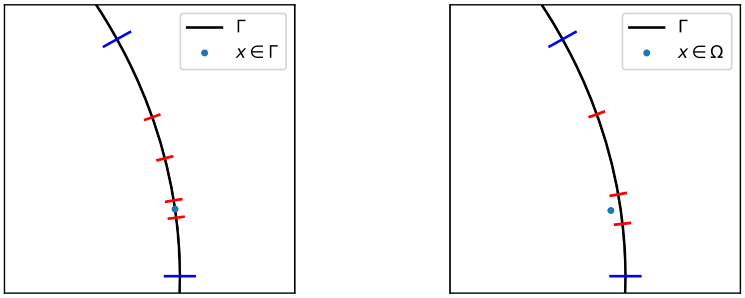


Figure 4: Illustration of the scheme used to subdivide the source panel when α is large, for a target point $\boldsymbol{x}$ either on or near the boundary Γ. The blue markings indicate the boundaries of the underlying source panel Γ, while the red markings indicate the boundaries of the new subpanels introduced when evaluating the layer potential at $\boldsymbol{x}$.

quadrature of section 4.3, which we use for the logarithmic kernel, works by rewriting a convolution integral as

$$\int f(z)K(\tau - z)\,\mathrm{d}\tau = \int f(z)K^S(\tau - z)\,\mathrm{d}\tau + \int f(z)K^L(\tau - z)\log(\tau - z)\,\mathrm{d}\tau, \tag{110}$$

and then evaluating the two integrals on the right hand side using Gauss-Legendre quadrature and complex interpolatory quadrature, respectively. Recall also that both quadratures rely on the remaining integrand ($f(z)K^S(\tau - z)$ and $f(z)K^L(\tau - z)$, respectively) being well approximated by a polynomial. Now, when K^S and K^L are of opposite sign, and both grow as $e^{\alpha|\tau - z|}$, this scheme will deteriorate quickly if $\alpha|\tau - z|$ gets large. This is due to a combination of two factors: interpolation errors, because the exponential factor is no longer well approximated by a polynomial of order $n - 1$, and catastrophic cancellation, due to the large magnitudes of the terms in the quadrature. This effectively introduces a maximum allowable value for the exponent $\alpha|\tau - z|$. Since kernel-split quadrature is only applied to target points z that are $\mathcal{O}(h)$ close to the source panel, this translates into a maximum allowable length of the source panels in the kernel-split quadrature. This can be expressed as the following criterion, which has been empirically determined for our problem,

$$\alpha h \leq 4.5. \tag{111}$$

While we could ensure that all panels satisfy the above criterion at discretization, this would force us to resolve the boundary more than necessary, and ultimately impose a time/space discretization restriction of the form $h^2/\delta t < C$ on our method. To avoid this, we use the scheme outlined in [6], which builds on the observation that the source panel Γ and the source density $\boldsymbol{\mu}$ are well resolved by discretization, and that the difficulties lie in properly resolving the kernel components $\mathcal{T}_1^S$–$\mathcal{T}_3^L$, which are analytically known functions. We can therefore subdivide the source panel Γ into a set of temporary subpanels, from which the contributions at z are evaluated, using boundary data interpolated from Γ. This allows us to choose the subpanels such that the ones requiring kernel-split quadrature, based on their relative distance from z, also satisfy (111). An efficient way of doing this is by using subpanels that are successively refined in the direction of the target point z, as illustrated in fig. 4. The resulting quadrature scheme is robust, and accurately evaluates the layer potential both for large α and target points very close to $\partial\Omega$. The hierarchical structure of the subdivision algorithm makes the scheme fast, even for large α. In addition, it only incurs an additional cost at the precomputation step; the action of the quadrature over the subpanels is reduced to a set of modified quadrature weights at the original 16 sources nodes, using interpolation matrices.

4.5.5. Identifying interior points

An important component of our method is knowing which of the grid points in B lie in Ω, as illustrated in fig. 1. There are many different ways of determining this, here we choose a layer potential approach. We first evaluate the Laplace double layer identity

$$\phi(\boldsymbol{x}) = -\frac{1}{2\pi}\int_{\partial\Omega} \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\hat{\boldsymbol{n}}}{\|\boldsymbol{x}-\boldsymbol{y}\|^2}\,\mathrm{d}S_{\boldsymbol{y}} = \begin{cases} 1, & \boldsymbol{x}\in\Omega, \\ 0, & \boldsymbol{x}\notin\Omega, \end{cases} \tag{112}$$

using an FMM and our underlying 16-point Gauss-Legendre quadrature. In the first pass, any point such that $\phi(\boldsymbol{x}) > \frac{1}{2}$ is marked as being interior. In the second pass, we use the algorithm of section 4.5.3 to find any points that are too close to $\partial\Omega$ for the quadrature approximation of (112) to be accurate. For these points, we find the closest boundary point $\boldsymbol{x}_c$ and mark them as interior if $(\boldsymbol{x}-\boldsymbol{x}_c)\cdot\hat{\boldsymbol{n}}(\boldsymbol{x}_c) > 0$. This can still fail if $\|\boldsymbol{x}-\boldsymbol{x}_c\|$ is smaller than the distance between $\boldsymbol{x}_c$ an its neighboring points on $\partial\Omega$. In such cases we compute the preimage of $\boldsymbol{x}$ under the parametrization of the nearest panel, as described in section 4.5.1, and mark the point as interior if $\mathrm{Im}[\gamma^{-1}(x_1+ix_2)] > 0$. This algorithm identifies interior points in a fast and robust fashion, and builds on methods that are already present in our code.

5. Numerical examples

5.1. Implementation

The software implementation of our scheme is for the most part written in Julia [14], and is available for reference as open source code [2]. The implementation depends on several external packages: For function extension, we use a Matlab implementation of PUX written by the authors of [26], available at [25]. This in turn makes use of the RBF-QR algorithm [23], as implemented in [46]. Non-uniform FFTs are computed using the library FINUFFT [12]. Our FMM implementation is written in Fortran, and is based on FMMLIB2D [29]. We also use FMMLIB2D directly for identifying interior points (see section 4.5.5). The fast direct solver is written in Julia, but is based on a Matlab implementation of the Stokes solver used in [48], provided by A. Gillman. The low rank matrix approximations used in the fast direct solver are computed using the interpolative decomposition (ID) [47, 20], as implemented in LowRankApprox.jl [41]. Our Julia code is single-threaded, though several of the libraries used, including BLAS operations, are multi-threaded.

The scheme is designed to be as efficient as possible in solving the inhomogeneous problem (9) for a fixed α, which is equivalent to taking IMEX or SISDC steps with a fixed time step size. To achieve this, we precompute the fast direct solver, the quadrature weights required to evaluate the layer potential at the near-boundary grid points, the FMM tree, and the main system matrix of PUX. This means that there is a significant cost associated with updating either the discretization or time step size. We therefore run all of our simulations with a constant time step size, and let SISDC have equisized substeps.

Throughout the below tests, we denote by $\boldsymbol{u}$ the reference solution, and by $\tilde{\boldsymbol{u}}$ the computed solution. For a discretization with grid points $\{\boldsymbol{x}_j\}_1^{N_\Omega}$, we measure the error $\boldsymbol{e} = \tilde{\boldsymbol{u}} - \boldsymbol{u}$ in the ℓ^2 and ℓ^∞ norms, computed as

$$\|\boldsymbol{e}\|_2 = \sqrt{\sum_{j=1}^{N_\Omega}\left(e_1(\boldsymbol{x}_j)^2 + e_2(\boldsymbol{x}_j)^2\right)} \quad \text{and} \quad \|\boldsymbol{e}\|_\infty = \max_{\substack{1\le j\le N_\Omega \\ i=1,2}} \left|e_i(\boldsymbol{x}_j)\right|. \tag{113}$$

Where reported, the computation times are from a computer with 32 GB of memory and an Intel Core i7-7700 CPU (4 physical cores, 3.6 GHz base frequency).

5.2. Homogeneous stationary problem

To validate the solver for the homogeneous equation (5), described in 3.2, we set up the following problem: The boundary $\partial\Omega$ is the starfish described by the parametrization

$$\boldsymbol{g}(t) = R\left(1 + A\cos 5t\right)\begin{pmatrix}\cos t \\ \sin t\end{pmatrix}, \quad t\in[0,2\pi), \tag{114}$$

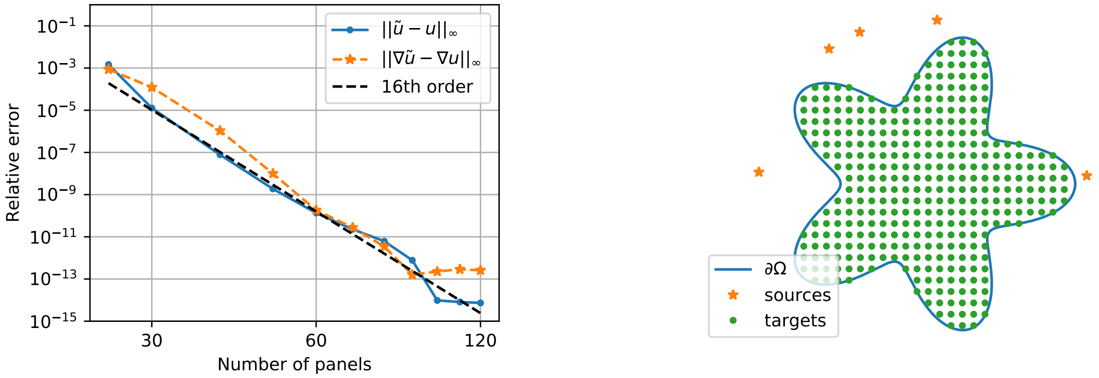


Figure 5: Convergence for the homogeneous solver, with $\alpha = 10$. The solution is given by stokeslets at the source points, and we measure the maximum relative errors in the solution and the gradient at the target points.

with $R = 1$ and $A = 0.3$. The Dirichlet boundary condition is given by a sum of 5 stokeslets,

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) = \sum_{n=1}^{5} S(\boldsymbol{x}, \boldsymbol{y}_n)\boldsymbol{q}_n, \quad \boldsymbol{x} \in \partial\Omega. \tag{115}$$

This has the exact solution $\boldsymbol{u} = \boldsymbol{f}$ in Ω. The source locations $\boldsymbol{y}_n$ are randomly placed on a circle of radius 1.4, and the strengths $\boldsymbol{q}_n$ are drawn randomly from $\mathcal{U}(-1, 1)$. We set $\alpha = 10$, and discretize $\partial\Omega$ using a varying number of panels, from 30 to 130. For each discretization, we first solve the integral equation by forming the dense linear system and solving it directly, and then evaluate the solution at 358 uniform target points in Ω using our fast summation and near-boundary quadrature. The geometry of the problem and the convergence results are shown in fig. 5. We expect to see a 16th order convergence, since that is the order of the Gauss-Legendre panels, and that is also what we observe, both in $\boldsymbol{u}$ and $\nabla\boldsymbol{u}$.

5.3. Inhomogeneous problem convergence

Next, we validate our composite solver for the full stationary problem — the inhomogeneous modified Stokes equations (9). We again use the starfish geometry (114), and let the right hand side $\boldsymbol{F}$ be a simple oscillation,

$$\boldsymbol{F}(\boldsymbol{x}) = (-1, 2)\cos(x_1 + x_2), \quad \boldsymbol{x} \in \Omega. \tag{116}$$

On a periodic box with sides 2π, this has an exact solution given by a single Fourier mode (25),

$$\boldsymbol{u}_{\text{per}}(\boldsymbol{x}) = (3, -3)\frac{\cos(x_1 + x_2)}{2(2 + \alpha^2)}, \quad \boldsymbol{x} \in \Omega. \tag{117}$$

By setting the Dirichlet boundary condition $\boldsymbol{u} = \boldsymbol{u}_{\text{per}}$ on $\partial\Omega$, we get a problem with exact solution $\boldsymbol{u} = \boldsymbol{u}_{\text{per}}$ in Ω. This tests the complete solver setup, since $\boldsymbol{F}$ is only given inside Ω, and then smoothly extended into the bounding box B using PUX (see fig. 7). We discretize $\partial\Omega$ using 400 panels, and set the PUX radius to $R = 0.15$. In Ω we use a uniform grid with $N \times N$ points in the smallest square box that bounds $\partial\Omega$.

To test convergence in the volume grid, we set $\alpha = 10$ and solve the above problem for a wide range of N, from 40 to 1300. The results, shown in fig. 6, indicate that we have 10th order convergence in both $\boldsymbol{u}$ and $\nabla\boldsymbol{u}$. This is in accordance with the results in [26]. Note that the errors differ by about two orders of magnitude between 2-norm and max-norm, and also by about two orders of magnitude between $\boldsymbol{u}$ and $\nabla\boldsymbol{u}$. A maximum relative error in $\nabla\boldsymbol{u}$ around 10^{-9} appears to be the best that we can achieve for this particular problem. This is due to errors in the solution close to the boundary, see fig. 7.

To test the dependence on the parameter α, defined as $\alpha^2 = Re/\delta t$, we also solve the above problem with α varying by several orders of magnitude. This time we keep N fixed at $N = 800$, where the solution

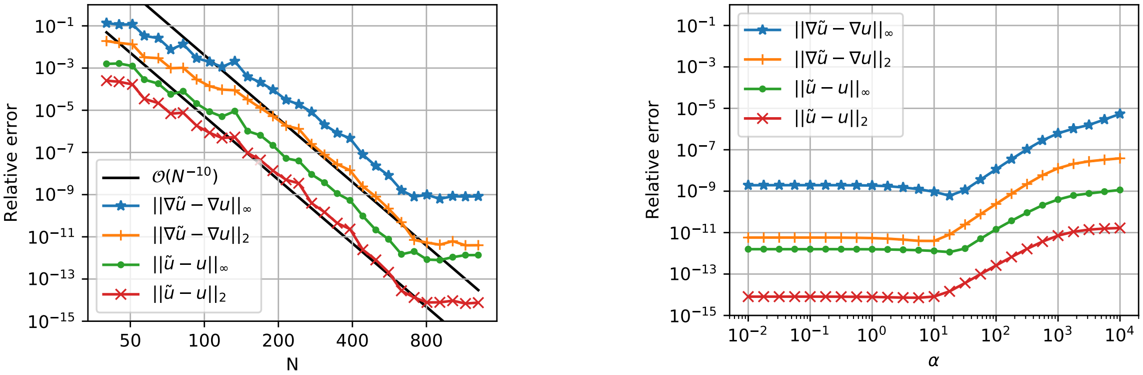


Figure 6: Left: Convergence for full problem, $\alpha = 10$. Right: Varying α when $N = 800$.

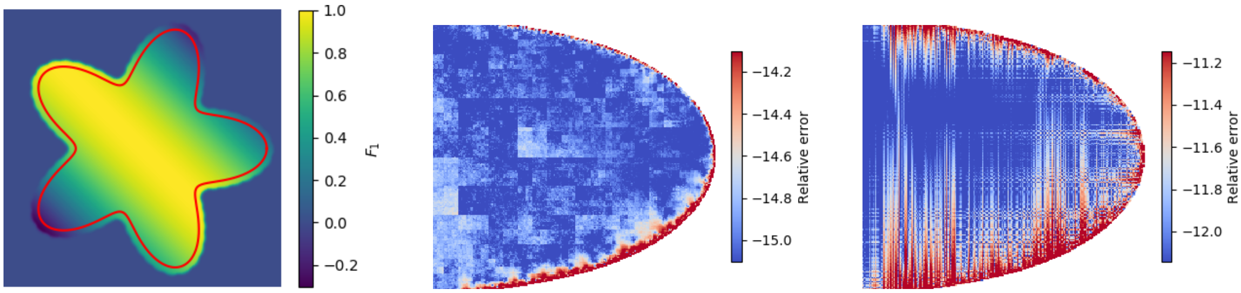


Figure 7: Left: Function extension of the first component of (116), using PUX with partition radius $R = 0.15$. Center and right: Zoom of the rightmost starfish arm, showing the error in $\boldsymbol{u}$ (pointwise maximum) for $N = 800$ and $\alpha = 1$ (center) and $\alpha = 1000$ (right). Note that the color scale is different between the two pictures. At $\alpha = 1$, the box-like structure in the error is due to the FMM tree. The error near the boundary is due to PUX; the scale is too large to be due to the layer potential near evaluation. The structure in the error for $\alpha = 1000$ is typical for errors in high Fourier modes, an effect of the reduced smoothing for large α.

was fully converged for $\alpha = 10$. The results, also shown in fig. 6, clearly show that accuracy suffers when α gets very large. We believe that this is error originates in the particular solution, as the homogeneous solver can handle a very wide range of α, due to the scheme outlined in section 4.5.4. Specifically, the smoothing effect of the Fourier multiplier (26) is reduced for large α, which exposes any irregularities in the function extensions. This problem is even more pronounced in the computation of the gradient (27), which helps explain the larger errors in $\nabla\boldsymbol{u}$.

5.4. Time convergence

To validate our complete solver for the Navier-Stokes equations (1), we consider the problem of viscous spin-down in a cylinder: A cylinder of radius a is filled with viscous fluid, and at $t = 0$ both the cylinder and the fluid are rotating with angular velocity Ω. Then (for $t > 0$) the cylinder suddenly stops rotating, while the fluid keeps rotating until it comes to rest. In polar coordinates,

$$\begin{aligned} \boldsymbol{u}(r,t) &= u_\theta(r,t)\hat{\boldsymbol{\theta}}, & & \\ u_\theta(r,0) &= \Omega r, & r &\le a, \\ u_\theta(a,t) &= 0, & t &> 0. \end{aligned} \tag{118}$$

The exact solution to this problem is known [1, p.45],

$$u_\theta(r,t) = -2\Omega a \sum_{n=1}^{\infty} \frac{J_1(\lambda_n r/a)}{\lambda_n J_0(\lambda_n)} e^{-\lambda_n^2 t/(a\,Re)}. \tag{119}$$

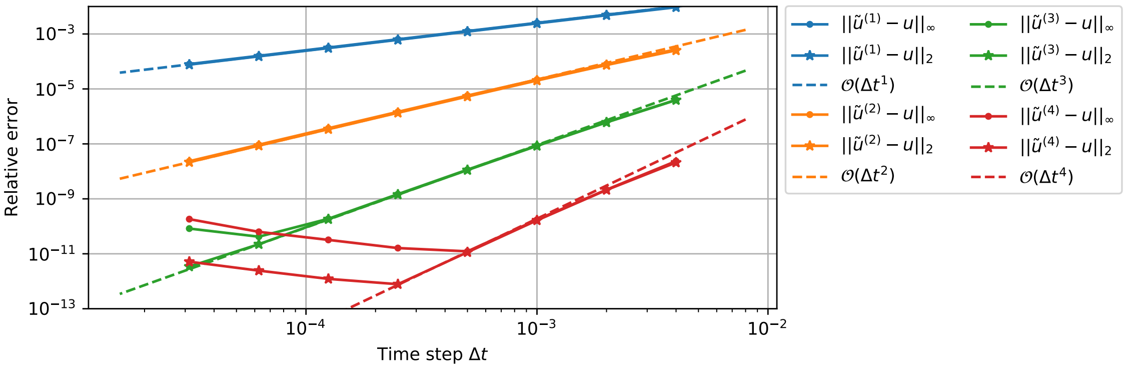


Figure 8: Time convergence for spin down problem, $\tilde{u}^{(K)}$ denotes solution computed using SISDC order K. For small Δt, the lowest attainable error can be seen to increase with decreasing Δt. This is related to the results in fig. 6, which show that the error grows with increasing α.

Here J_k is the Bessel function of order k, and λ_n is the nth positive root of J_1.

We solve the above problem for $a = \Omega = Re = 1$. In order to avoid the time-discontinuity in the boundary condition at $t = 0$, we initialize our solver at $t = 0.05$. The flow field is then sufficiently smooth, so that we can observe high-order convergence. We run our simulation on the interval $t \in [0.05, 0.054]$, with an initial time step length $\Delta t = 0.002$, which is then successively halved. Our discretization uses a 500×500 volume grid, 200 panels on the boundary, and PUX radius $R = 0.3$. We time step using SISDC of order 1 through 4, and compare the results in fig. 8. Convergence is as expected in both ℓ^2 and ℓ^∞, although the fourth order convergence only just reaches the expected rate before being overtaken by what appears to be an error that grows with decreasing Δt, and is approximately the same for $K = 3$ and $K = 4$. To explain this, recall that with time step length Δt and SISDC order K, the IMEX substep length is $\delta t = \Delta t/(K-1)$, such that

$$\alpha = \sqrt{\frac{(K-1)\,Re}{\Delta t}}. \tag{120}$$

For this time convergence test, the smallest time steps used correspond to values α in the region where the errors in the stationary solver start increasing, as seen in fig. 6. This explains the increasing error, and also why it is approximately the same between $K = 3$ and $K = 4$.

5.5. Stability

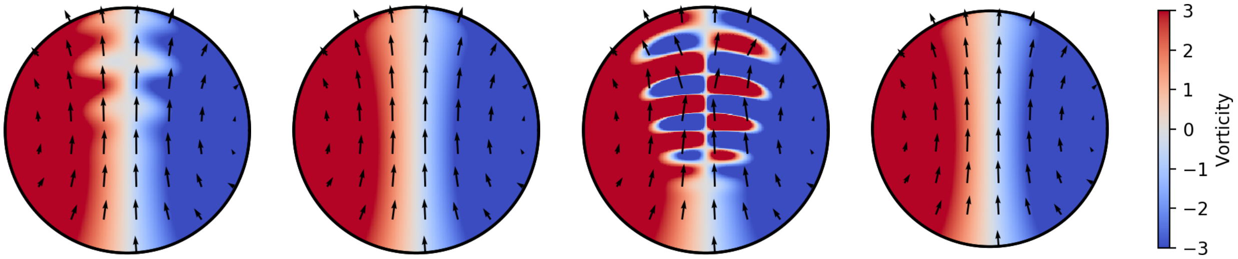
To investigate the stability properties of our method, we consider the simple case of flow in a circle of diameter 1, with boundary condition

$$\boldsymbol{u} = -\sin(\phi)\hat{\boldsymbol{n}}, \quad \boldsymbol{x} \in \partial\Omega, \tag{121}$$

where $\phi = \tan^{-1}(x_2/x_1)$ is the angle of the point on the boundary. Simulating this flow with the IMEX scheme, we observe instabilities in the form of waves in the vorticity, originating at the inflow, as shown in fig. 9. Through numerical experimentation, we find that these instabilities occur unless we satisfy a condition of the form

$$\delta t\, Re \leq C, \tag{122}$$

where, for this particular flow problem, $C \approx 1$. The instabilities do not appear to depend on the spatial scale (radius of the circle), nor the spatial discretization. The examples in fig. 9 were computed using a high-resolution spatial discretization, but refining or coarsening that discretization has no effect, as long as it is sufficiently fine for accurately solving the modified Stokes equation. Switching to higher-order SISDC is stabilizing, effectively increasing C, but does not change the form of the stability condition.



(a)
$\delta t = 0.04$
$Re = 50$

(b)
$\delta t = 0.02$
$Re = 50$

(c)
$\delta t = 0.02$
$Re = 100$

(d)
$\delta t = 0.01$
$Re = 100$

Figure 9: Simulations of the flow (121), with varying δt and Re. Cases (b) and (d) satisfy the heuristic stability condition (122), and develop a steady state. Snapshots shown are after 200 time steps. On the other hand, cases (a) and (c) violate the stability condition, and develop an instability pattern that is more pronounced for larger Re. The snapshots shown are after 45 time steps; the solution eventually blows up completely if run longer.

As an explanation model for our observations, we consider the standard stability analysis of the 1D advection-diffusion equation, which we write as

$$u_t + u_x - \frac{1}{Re} u_{xx} = 0. \tag{123}$$

This serves as a linearized model of the non-dimensionalized Navier-Stokes equations. We discretize this using first-order IMEX in time and a Fourier series in space, writing $u(x, t_n) = \sum_k \hat{u}_k^n e^{ikx}$. This is representative of how we solve the particular problem (4), and diagonalizes our model problem to the difference equation

$$\frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{\delta t} + ik\hat{u}_k^n + \frac{1}{Re} k^2 \hat{u}_k^{n+1} = 0. \tag{124}$$

For this to have a stable solution, it must hold that $|\hat{u}_k^{n+1}| \leq |\hat{u}_k^n|$ for all k. It is straightforward to show that this leads to the condition

$$\delta t \, Re \leq 2. \tag{125}$$

This is consistent with our observed condition (122). Although derived using a simplified model, it is a strong indication of the origin of the observed instability.

Considering the physics of our problem, our results are perhaps not surprising. The smaller the viscosity is compared to the velocity, the more advection dominated the flow is. It is then only natural that the timestep must be small for the solution to be stable when the advective term is treated explicitly.

5.6. Flow past obstacles

As a demonstration of flow through a relatively complex geometry, we set up the geometry shown in fig. 10. The outer boundary is the box-like domain described by the parametrization

$$\boldsymbol{g}(t) = \frac{(W \cos t, H \sin t)}{\big((\cos t)^p + (\sin t)^p\big)^{1/p}}, \quad t \in [0, 2\pi), \tag{126}$$

with $(W, H) = (5, 3)$ and $p = 10$. The eight starfish-shaped inclusions are rotated and translated instances of the curve given by (114), with $R = 0.5$ and $A = 0.2$. The boundary conditions are slip on the outer boundary, $\boldsymbol{u} = (1, 0)$, and no-slip on the inclusions, $\boldsymbol{u} = (0, 0)$. The Reynolds number is set to $Re = 30$.

The problem is discretized using a grid of 500 × 300 points in the domain, 500 panels on the outer boundary, 50 panels on each inclusion, PUX radius 0.4, and second order SISDC with $\Delta t = 0.01$. This

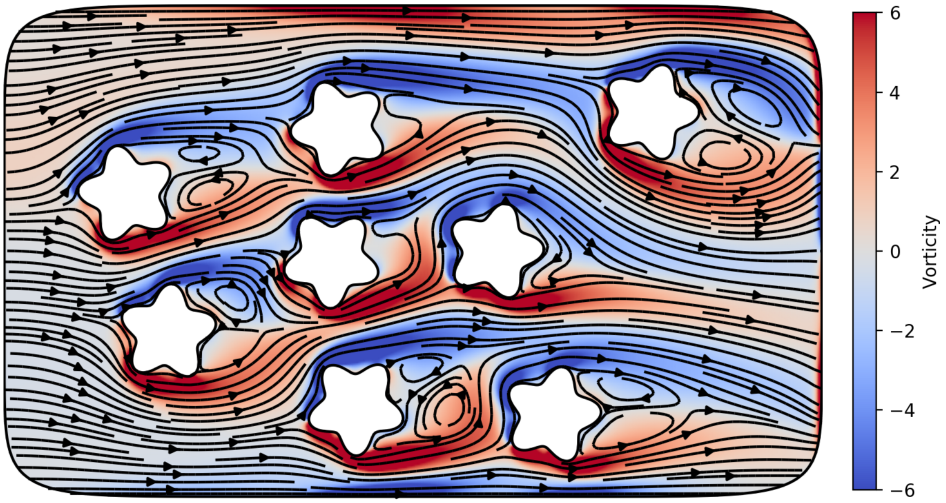


Figure 10: Streamlines and vorticity field of flow past a collection of starfish at $Re = 30$.

gives $\alpha = 54.8$. The homogeneous problem is solved using the fast direct solver (FDS), computed using a tolerance of 10^{-8}. This spatial discretization gives an error of $\|\boldsymbol{u} - \tilde{\boldsymbol{u}}\|_\infty = 5 \cdot 10^{-6}$ on the test problem used in section 5.3.

The precomputation time for this problem is dominated by the FDS, which takes 160 s to compute. Remaining precomputations that take more than one second are 6 s for PUX, and 10 s for the nearly singular quadrature. Once everything is precomputed, each solve takes around 2.5 s, where the average times for the involved algorithms are PUX: 0.5, FFT+NUFFT: 0.15, FDS: 0.03, FMM: 1.7. With second order SISDC (2 solves per step), each time step takes around 5 s.

We run the simulation for 10000 steps, after which the flow has reached a steady state. In fig. 10, we plot the streamlines and the vorticity, which is defined as

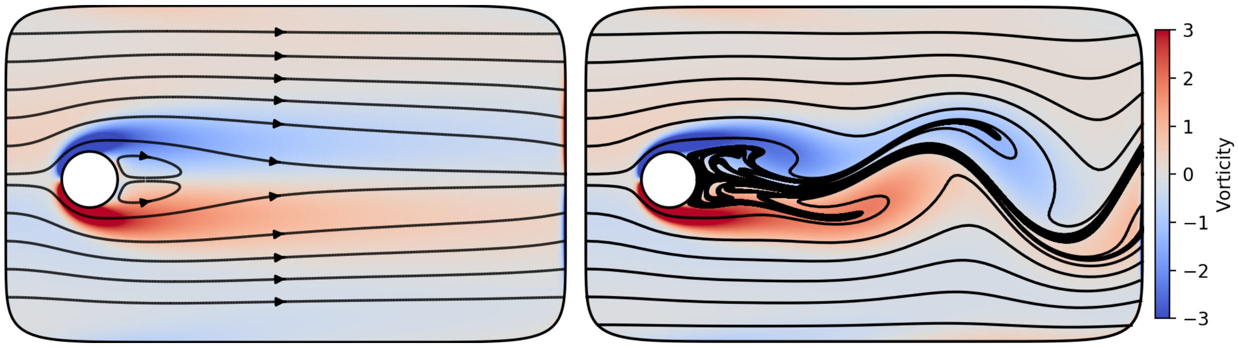
$$\omega = \frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y} \,. \tag{127}$$

5.7. *Vortex shedding*

To demonstrate the effect of the Reynolds number on time-dependent flow, we consider the case of flow past a cylinder, shown in figs. 11 and 12. For low values of the Reynolds number, the flow past a cylinder is stationary and symmetric, forming two standing eddies in the wake behind the cylinder (visible in fig. 11a). For values of the Reynolds number above a critical threshold value, the eddies begin to oscillate and detach, forming an oscillating tail of trailing vortices (visible in figs. 11b, 12a and 12b). This is known as a von Kármán vortex street [1].

For our simulation we let $Re = \{25, 50, 100, 200\}$. We use an outer domain that has the same box-shaped domain as in section 5.6, with the same slip boundary condition, and add an inclusion in the form of a circle with diameter 1 and a no-slip boundary condition. The circle is centered at $(-3.5, -0.1)$; the vertical offset is added because it reduces the number of time steps required before the vortex separation occurs.

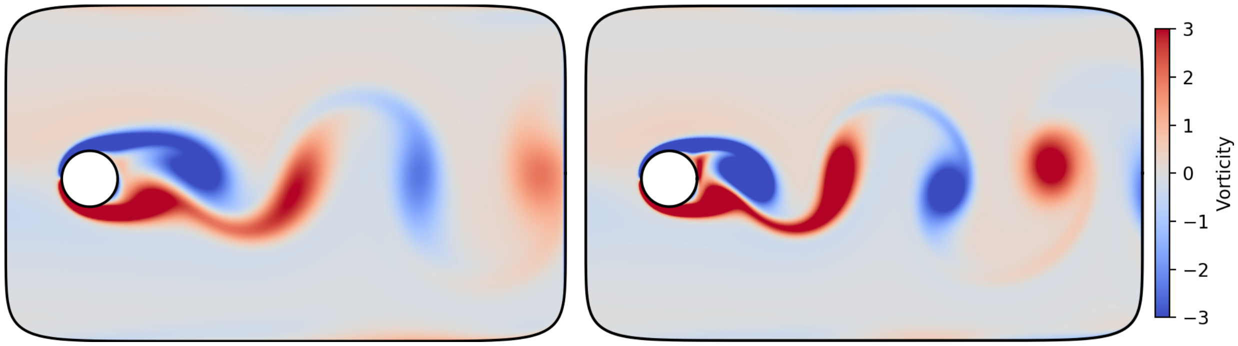
We discretize the problem using 500 panels on the outer boundary, and 100 panels on the circle. The domain is discretized using 500×300 points and PUX radius 0.5 for all values of Re except $Re = 200$, which is discretized using 1000×600 points and PUX radius 0.25, in order to get high accuracy for the larger α. We time step using second order SISDC, and observe that we need to set $\Delta t = 1/\,Re$ for the solution to be stable. This is consistent with the results in section 5.5, and as a result $\alpha = Re$. Again using the test problem of section 5.3 for reference, these discretizations have an accuracy of $\|\boldsymbol{u} - \tilde{\boldsymbol{u}}\|_\infty < 10^{-5}$ for all four values of α. Each solution is timestepped until it is either steady, or the dynamics of the unsteady solution



(a) $Re = 25$

(b) $Re = 50$

Figure 11: Flow past cylinder, with streaklines drawn in black and field colored by vorticity. At $Re = 25$ the flow is steady with standing eddies in the wake, while at $Re = 50$ the flow is unsteady with a vortex street in the wake.



(a) $Re = 100$

(b) $Re = 200$

Figure 12: Unsteady flow past cylinder, with a vortex street in the wake. Field colored by vorticity.

are fully developed. For $Re \geq 50$ we plot snapshots of the solutions at times such that all three solutions look qualitatively similar, for ease of comparison.

The flow behind a cylinder in a free stream develops a vortex street for Reynolds numbers above approximately 40. This is consistent with our simulation, which is steady for $Re = 25$, and develops a vortex street for $Re = 50$ and higher. Even though it is confined to a relatively small domain, with free-stream boundary conditions at the outer boundary, the results in figs. 11 and 12 match qualitatively with other results in the literature, e.g. [60].

6. Conclusion

In this work we have demonstrated a fast integral equation method (FIEM) for the Navier–Stokes equations in general smooth geometries in two dimensions. The solver has a number of favorable features. In terms of accuracy, the solver is 10th-order accurate in the spatial grid spacing, and we have demonstrated up to 4th-order convergence in time. By using an embedded boundary approach combined with a uniform grid, the method can effortlessly deal with complex geometries. In addition, the use of potential methods means that the solution satisfies the incompressibility constraint by construction. This eliminates the need for projection methods, and the artificial boundary conditions associated with them. It also makes it straightforward to use off-the-shelf methods to augment the temporal accuracy of the underlying IMEX discretization, and one could easily replace the SISDC scheme with, for example, a linear multistep method based on a backward differentiation formula (BDF) discretization.

Our numerical results indicate that our method can be used to solve the unsteady Navier-Stokes equations to high accuracy, for Reynolds numbers up to at least $Re = 200$. They do however also point towards a fundamental limitation in using a discretization based on Rothe's method. For large values of α (corresponding to large Reynolds numbers or short time steps), we see that more and more spatial resolution is needed to resolve the underlying kernels; see for instance, fig. 6. This is due to the modified Stokes equation being fundamentally harder to solve for large α. Combined with the stability restriction (122), which implies the parameter scaling $\alpha \sim Re$, this leads us to believe that our method will mainly be found useful for flows with low or moderate Reynolds numbers, in the range of hundreds up to a thousand.

For computations in two dimensions, there are several straightforward directions in which the present method can be developed. The integral equation formulation can be extended to allow for a wider class of boundary conditions, and to allow post-solution computation of various quantities, such as the pressure. While our method applies only to smooth domains, there has been recent progress in the efficient discretization of boundary integral equations on domains with corners, see, for instance, [58, 56, 40, 38]. Implementing one of these schemes would allow accurate computation of flows around geometries with sharp corners. It would also be of interest to extend our method to deal with moving geometries, such as drops, bubbles, vesicles, and particles. Here, the primary challenge would be to achieve high order accuracy while using function extension for computing the volume potential. A secondary challenge would be to maintain the efficiency of the method, as it currently relies heavily on precomputations for static geometries (fast direct solver, quadrature weights). For a flow such as bubbles in a channel, where there is a mix of moving and static boundaries, the precomputations would be limited to the static walls, as in [48]. The homogeneous problem for the boundary correction would then have to be solved iteratively, making preconditioning a factor to consider. Alternatively, for slow deformations it might be possible to use a fast direct solver combined with low rank updates [65].

Although there are several ways in which the present work may be improved, we believe that the most interesting direction of development is to extend it to three dimensions (3D). Many of the tools used in this work extend directly to 3D, but not all. In particular, the panel-based, kernel-split quadrature is limited to 2D, but could be replaced by the quadrature-by-expansion (QBX) method [44, 61]. In addition, high performance computing aspects become much more important in 3D, due to the large amount of degrees of freedom, and issues of adaptivity and parallelization must be considered from the start.

Acknowledgments

The authors gratefully acknowledge support from the Knut and Alice Wallenberg Foundation under grant no. 2016.0410 (LaK), from the Air Force Office of Scientific Research under grant FA9550-17-1-0329 (TA), and from the Natural Science and Engineering Research Council of Canada (MCK). We also wish to thank Adrianna Gillman for providing us with an implementation of a fast direct solver, and Fredrik Fryklund for providing us with an implementation of PUX.

Appendix A Modified Stokes layer potentials

In this section, we provide some technical results which describe the properties of the modified Stokes double layer potential and which explain the use of the nullspace correction. We closely follow the presentation of [15]. Details and proofs may be found in [43] for the Stokes case and [55, 15] for the modified Stokes case.

For reference, the homogeneous modified Stokes equation with the Dirichlet boundary condition is given by

$$(\alpha^2 - \Delta)\boldsymbol{v} + \nabla p = 0, \qquad \boldsymbol{x} \in \Omega, \tag{A.1}$$

$$\nabla \cdot \boldsymbol{v} = 0, \qquad \boldsymbol{x} \in \Omega, \tag{A.2}$$

$$\boldsymbol{v} = \boldsymbol{g}, \qquad \boldsymbol{x} \in \partial\Omega\ . \tag{A.3}$$

As in the main text, the divergence-free condition implies that the boundary data should satisfy

$$\int_\Omega \boldsymbol{g} \cdot \hat{\boldsymbol{n}}\ \mathrm{d}S_{\boldsymbol{y}}\ . \tag{A.4}$$

The following energy-type result is convenient for establishing the uniqueness of various boundary value problems for the modified Stokes equation.

Lemma 1 (Energy [15])**.** *Suppose that* $\boldsymbol{v}$ *satisfies* (A.1) *and* (A.2) *on a (possibly multiply-connected) domain* Ω *and let* $\boldsymbol{\sigma}(\boldsymbol{v})$ *denote the associated stress tensor. Then*

$$\int_\Omega \alpha^2 |\boldsymbol{v}|^2 + \frac{1}{2}\|\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^\intercal\|_F^2\ \mathrm{d}V_{\boldsymbol{y}} = -\int_{\partial\Omega} \big(\boldsymbol{\sigma}(\boldsymbol{v}) \cdot \hat{\boldsymbol{n}}\big) \cdot \boldsymbol{v}\ \mathrm{d}S_{\boldsymbol{y}}\ . \tag{A.5}$$

The quantity $-\boldsymbol{\sigma}(\boldsymbol{v}) \cdot \hat{\boldsymbol{n}}$ is sometimes referred to as the surface traction. For the modified Stokes equation, specifying the surface traction is the natural Neumann boundary value problem. We have

Corollary 1 (Uniqueness [15])**.** *Let* Ω *be a bounded (possibly multiply-connected) domain. There is at most one solution (up to an additive constant on p) of* (A.1) *and* (A.2) *specifying either the velocity field,* $\boldsymbol{v}$*, or the surface traction,* $\boldsymbol{\sigma}(\boldsymbol{v}) \cdot \hat{\boldsymbol{n}}$*, on* $\partial\Omega$*.*

Corollary 2 (Uniqueness, exterior [15])**.** *Let* Ω *be the exterior of a finite collection of bounded, simply-connected domains. There is at most one solution (up to an additive constant on p) of* (A.1) *and* (A.2) *which satisfies* $|\boldsymbol{v}| = o(1/|\boldsymbol{x}|)$ *as* $|\boldsymbol{x}| \to \infty$*, specifying either the velocity field,* $\boldsymbol{v}$*, or the surface traction,* $\boldsymbol{\sigma}(\boldsymbol{v}) \cdot \hat{\boldsymbol{n}}$*, on* $\partial\Omega$*.*

While it was not necessary to the discussion in the main text, the modified Stokes single layer potential is useful in the discussion of jump conditions and invertibility. For a given single layer density $\boldsymbol{\mu}$ defined on $\partial\Omega$, the potential is denoted by $\mathcal{S}[\boldsymbol{\mu}]$ and is defined by the boundary integral

$$\mathcal{S}_i[\boldsymbol{\mu}](\boldsymbol{x}) = \int_{\partial\Omega} S_{ij}(\boldsymbol{x}, \boldsymbol{y}) \mu_j(\boldsymbol{y})\, \mathrm{d}S_{\boldsymbol{y}}, \tag{A.6}$$

where S_{ij} is the Stokeslet as defined in (11). Denote the stress tensor associated with the single layer potential by $\boldsymbol{\sigma}_{\mathcal{S}}[\boldsymbol{\mu}]$. Another layer potential of interest corresponds to the surface traction of the single layer potential. We denote this layer potential by $\mathcal{N}[\boldsymbol{\mu}]$, and for $\boldsymbol{x} \in \partial\Omega$, it is defined by

$$\mathcal{N}[\boldsymbol{\mu}]_i(\boldsymbol{x}) = -\oint_{\partial\Omega} T_{ijk}(\boldsymbol{x}, \boldsymbol{y}) \mu_j(\boldsymbol{y}) \hat{n}_k(\boldsymbol{x})\, \mathrm{d}S_{\boldsymbol{y}}, \tag{A.7}$$

where the $\oint$ symbol indicates that this integral is interpreted in the Cauchy principal value sense. Note that this integral operator is the transpose of the double layer operator.

The modified Stokes single and double layer potentials satisfy jump conditions which are analogous to the more familiar jump conditions for harmonic layer potentials.

Lemma 2 (Jump conditions)**.** *Let $\mathcal{S}$, $\boldsymbol{\sigma}_{\mathcal{S}}$, $\mathcal{N}$, and $\mathcal{D}$ be the layer potentials as defined above and let $\partial\Omega$ be a sufficiently smooth domain boundary with inward pointing normal $\hat{\boldsymbol{n}}$. Then, for a given density $\boldsymbol{\mu}$ defined on $\partial\Omega$, we have that $\mathcal{S}\boldsymbol{\mu}$ is continuous across $\partial\Omega$, the exterior and interior limits of the surface traction of $\mathcal{D}\boldsymbol{\mu}$ are equal, and for each $\boldsymbol{x}_0 \in \partial\Omega$,*

$$\lim_{h\to 0} -\boldsymbol{\sigma}_{\mathcal{S}}[\boldsymbol{\mu}](\boldsymbol{x}_0 \pm h\hat{\boldsymbol{n}}(\boldsymbol{x}_0)) \cdot \hat{\boldsymbol{n}}(\boldsymbol{x}_0) = \mp\frac{1}{2}\boldsymbol{\mu}(\boldsymbol{x}_0) + \mathcal{N}[\boldsymbol{\mu}](\boldsymbol{x}_0) \tag{A.8}$$

$$\lim_{h\to 0} \mathcal{D}[\boldsymbol{\mu}](\boldsymbol{x}_0 \pm h\hat{\boldsymbol{n}}(\boldsymbol{x}_0)) = \pm\frac{1}{2}\boldsymbol{\mu}(\boldsymbol{x}_0) + \mathcal{D}[\boldsymbol{\mu}](\boldsymbol{x}_0)\ . \tag{A.9}$$

Note that the integral in the definition of $\mathcal{D}[\boldsymbol{\mu}](\boldsymbol{x}_0)$ is interpreted in the Cauchy principal value sense when $\boldsymbol{x}_0 \in \partial\Omega$.

The above expressions are derived by noting that the leading order singularity of these integral kernels is the same as for the original Stokes case, so that the standard jump conditions for Stokes [43, 55] apply. The correspondence between the $\mathcal{D}$ and $\mathcal{N}$ operators, namely that $\mathcal{D} = \mathcal{N}^{\intercal}$, helps to characterize the nullspace of $\frac{1}{2} + \mathcal{D}$.

Lemma 3 (Nullspace [15])**.** *Suppose that $\partial\Omega$ is the boundary of a (possibly multiply-connected) bounded domain and that $\hat{\boldsymbol{n}}$ denotes the inward pointing normal. Then $\dim(N(\frac{1}{2} + \mathcal{D})) = 1$. If $\boldsymbol{\xi} \in N(\frac{1}{2} + \mathcal{D})$ and $\boldsymbol{\xi} \neq \boldsymbol{0}$, then $\int_{\partial\Omega} \boldsymbol{\xi} \cdot \hat{\boldsymbol{n}} \neq 0$.*

In order to deal with the rank 1 nullspace described above, we add the term

$$\mathcal{W}[\boldsymbol{\mu}](\boldsymbol{x}) = \frac{\hat{\boldsymbol{n}}(\boldsymbol{x})}{\int_{\partial\Omega} \mathrm{d}S} \int_{\partial\Omega} \boldsymbol{\mu}(\boldsymbol{y}) \cdot \hat{\boldsymbol{n}}(\boldsymbol{y})\, \mathrm{d}S_{\boldsymbol{y}} \tag{A.10}$$

to the double layer potential. This process is sometimes referred to as a nullspace correction or Wielandts deflation [15, 24]. We summarize this process in

Lemma 4 (Invertibility)**.** *Let $\mathcal{W}$ and $\mathcal{D}$ be defined as above. Then, the equation*

$$\frac{1}{2}\boldsymbol{\mu} + \mathcal{D}[\boldsymbol{\mu}] + \mathcal{W}[\boldsymbol{\mu}] = \boldsymbol{f}, \quad \boldsymbol{x} \in \partial\Omega \tag{A.11}$$

is uniquely invertible. Moreover, $\mathcal{W}[\boldsymbol{\mu}]$ is zero provided that the right hand side satisfies the condition

$$\int_{\partial\Omega} \boldsymbol{f} \cdot \hat{\boldsymbol{n}}\ \mathrm{d}S_{\boldsymbol{y}} = 0\ . \tag{A.12}$$

Proof. Suppose that $\boldsymbol{f}$ satisfies (A.12). Taking the inner product of (A.11) with $\hat{\boldsymbol{n}}$ and integrating, we obtain

$$\int_{\partial\Omega} \hat{\boldsymbol{n}} \cdot \left(\frac{1}{2} + \mathcal{D} + \mathcal{W}\right)\boldsymbol{\mu} = 0\ . \tag{A.13}$$

Because $\boldsymbol{u} = \mathcal{D}\boldsymbol{\mu}$ is divergence-free and $(\frac{1}{2} + \mathcal{D})\boldsymbol{\mu}$ gives $\boldsymbol{u}$ on the boundary, the above implies that $\int \boldsymbol{\mu} \cdot \hat{\boldsymbol{n}} = 0$ and $\mathcal{W}\boldsymbol{\mu} = \boldsymbol{0}$

By the Fredholm alternative, it is sufficient to show that

$$\frac{1}{2}\boldsymbol{\mu} + \mathcal{D}[\boldsymbol{\mu}] + \mathcal{W}[\boldsymbol{\mu}] = \boldsymbol{0}, \quad \boldsymbol{x} \in \partial\Omega \tag{A.14}$$

implies that $\boldsymbol{\mu} = \boldsymbol{0}$. Note that $\boldsymbol{f} = \boldsymbol{0}$ certainly satisfies (A.12). Therefore, $\int \boldsymbol{\mu} \cdot \hat{\boldsymbol{n}} = 0$ and $(\frac{1}{2} + \mathcal{D})\boldsymbol{\mu} = \boldsymbol{0}$. Then, by Lemma 3, we have that $\boldsymbol{\mu} = \boldsymbol{0}$.

□

Appendix B Free-space Green's functions of the modified Stokes equations

We here derive the fundamental solutions to the modified Stokes equations needed for our application. Our starting point is the fundamental solution to the modified biharmonic equation,

$$\Delta(\Delta - \alpha^2)G(\boldsymbol{r}) = \delta(\boldsymbol{r}). \tag{B.1}$$

In two dimensions [42],

$$G(\boldsymbol{r}) = -\frac{1}{2\pi\alpha^2}\left(\log r + K_0(\alpha r)\right), \tag{B.2}$$

where $r = \|\boldsymbol{r}\|$ and K_n is a modified Bessel functions of the second kind. We now seek the fundamental solution to the modified Stokes equation, such that

$$\alpha^2\boldsymbol{u}(\boldsymbol{r}) - \Delta\boldsymbol{u}(\boldsymbol{r}) + \nabla p(\boldsymbol{r}) = \delta(\boldsymbol{r})\boldsymbol{f}, \tag{B.3}$$

$$\nabla \cdot \boldsymbol{u}(\boldsymbol{r}) = 0, \tag{B.4}$$

where $\boldsymbol{f}$ is an arbitrary constant. Substituting $\delta(\boldsymbol{r})$ using (B.1) and taking the divergence,

$$\Delta p = \nabla \cdot \Delta(\Delta - \alpha^2)G\boldsymbol{f}, \tag{B.5}$$

such that

$$p = (\Delta - \alpha^2)\nabla G \cdot \boldsymbol{f}. \tag{B.6}$$

Substituting (B.1) and (B.6) into (B.3),

$$(\alpha^2 - \Delta)\boldsymbol{u} = (\alpha^2 - \Delta)\nabla(\nabla G \cdot \boldsymbol{f}) + \Delta(\Delta - \alpha^2)G\boldsymbol{f}, \tag{B.7}$$

$$= (\alpha^2 - \Delta)(\nabla \otimes \nabla)G\boldsymbol{f} - (\alpha^2 - \Delta)\Delta G\boldsymbol{f}, \tag{B.8}$$

$$\boldsymbol{u} = (\nabla \otimes \nabla - \Delta I)\, G\boldsymbol{f}, \tag{B.9}$$

such that the stokeslet tensor S is given by

$$S(\boldsymbol{r}) = (\nabla \otimes \nabla - \Delta)G(\boldsymbol{r}). \tag{B.10}$$

We now switch to index notation, following the Einstein summation convention and using the notation $\partial_i = \frac{\partial}{\partial r_i}$. The Kronecker delta δ_{ij} is not to be confused with the Dirac delta function $\delta(\boldsymbol{r})$ used above. The stokeslet is then written

$$S_{ij}(\boldsymbol{r}) = \left(\partial_i\partial_j - \Delta\delta_{ij}\right) G(\boldsymbol{r}), \tag{B.11}$$

and the corresponding pressure vector is given by

$$\phi_i(\boldsymbol{r}) = \partial_i(\Delta - \alpha^2)G(\boldsymbol{r}). \tag{B.12}$$

The stresslet tensor, defined as

$$T_{ijk}(\boldsymbol{r}) = -\delta_{ik}\phi_j(\boldsymbol{r}) + \partial_k S_{ij}(\boldsymbol{r}) + \partial_i S_{kj}(\boldsymbol{r}), \tag{B.13}$$

is then given by

$$T_{ijk} = -\left(\delta_{ij}\partial_k + \delta_{jk}\partial_i + \delta_{ki}\partial_j\right)\Delta G + 2\partial_i\partial_j\partial_k G + \alpha^2\delta_{ik}\partial_j G. \tag{B.14}$$

Note that for $\alpha = 0$ we recover the corresponding tensors for regular Stokes flow, with G being the fundamental solution to the biharmonic equation.

We have that G is a radial function in r, $G(\boldsymbol{r}) = G(r)$. Using that in dimension d

$$\partial_i r = \frac{r_i}{r}, \qquad \partial_i r_j = \delta_{ij}, \qquad \partial_i r_i = d, \qquad r_i r_i = r^2, \tag{B.15}$$

and

$$\Delta G(\boldsymbol{r}) = G''(\boldsymbol{r}) + (d-1)\frac{G'(r)}{r}, \tag{B.16}$$

it is straightforward to derive that

$$\partial_i G(\boldsymbol{r}) = G'(\boldsymbol{r})\frac{r_i}{r}, \tag{B.17}$$

$$\partial_i \partial_j G(\boldsymbol{r}) = \frac{G'(\boldsymbol{r})}{r}\delta_{ij} + \left(G''(\boldsymbol{r}) - \frac{G'(\boldsymbol{r})}{r}\right)\frac{r_i r_j}{r^2}, \tag{B.18}$$

$$\begin{aligned}\partial_i \partial_j \partial_k G(\boldsymbol{r}) = {} & \left(\frac{G''(r)}{r} - \frac{G'(r)}{r^2}\right)\frac{\delta_{jk} r_i + \delta_{ik} r_j + \delta_{ij} r_k}{r} \\ & + \left(G'''(r) - 3\frac{G''(r)}{r} + 3\frac{G'(r)}{r^2}\right)\frac{r_i r_j r_k}{r^3},\end{aligned} \tag{B.19}$$

$$\partial_i \Delta G(\boldsymbol{r}) = \left(G'''(\boldsymbol{r}) + (d-1)\frac{G''(\boldsymbol{r})}{r} - (d-1)\frac{G'(\boldsymbol{r})}{r^2}\right)\frac{r_i}{r}. \tag{B.20}$$

Using these relations, we can now write down S, $\boldsymbol{\phi}$ and T in terms of G, G' and G'',

$$S_{ij}(\boldsymbol{r}) = -\frac{1}{r}\left(rG''(\boldsymbol{r}) + (d-2)G'(\boldsymbol{r})\right)\delta_{ij} + \frac{1}{r^3}\left(rG''(\boldsymbol{r}) - G'(\boldsymbol{r})\right) r_i r_j, \tag{B.21}$$

$$\phi_i(\boldsymbol{r}) = \frac{1}{r^3}\left(r^2 G'''(\boldsymbol{r}) + (d-1) r G''(\boldsymbol{r}) - \left(r^2\alpha^2 + d - 1\right) G'(\boldsymbol{r})\right) r_i, \tag{B.22}$$

$$\begin{aligned}T_{ijk}(\boldsymbol{r}) = {} & \frac{1}{r^3}\left(-r^2 G'''(\boldsymbol{r}) + (3-d) r G''(\boldsymbol{r}) - (3-d) G'(\boldsymbol{r})\right)\left(\delta_{jk} r_i + \delta_{ik} r_j + \delta_{ij} r_k\right) \\ & + \frac{2}{r^5}\left(r^2 G'''(\boldsymbol{r}) - 3 r G''(\boldsymbol{r}) + 3 G'(\boldsymbol{r})\right) r_i r_j r_k + \frac{\alpha^2 G'(\boldsymbol{r})}{r}\delta_{ik} r_j.\end{aligned} \tag{B.23}$$

In two dimensions (using $d = 2$ and (B.2)) we can write

$$S_{ij}(\boldsymbol{r}) = \mathcal{S}_1(\alpha r)\delta_{ij} + \alpha^2 \mathcal{S}_2(\alpha r) r_i r_j, \tag{B.24}$$

$$\phi_i(\boldsymbol{r}) = \frac{r_i}{2\pi r^2}, \tag{B.25}$$

$$T_{ijk}(\boldsymbol{r}) = \alpha^2 \mathcal{T}_1(\alpha r)\left(\delta_{jk} r_i + \delta_{ik} r_j + \delta_{ij} r_k\right) + \alpha^4 \mathcal{T}_2(\alpha r) r_i r_j r_k + \alpha^2 \mathcal{T}_3(\alpha r)\delta_{ik} r_j, \tag{B.26}$$

where

$$\mathcal{S}_1(z) = \frac{z^2 K_0(z) + z K_1(z) - 1}{2\pi z^2}, \tag{B.27}$$

$$\mathcal{S}_2(z) = -\frac{z^2 K_0(z) + 2z K_1(z) - 2}{2\pi z^4}, \tag{B.28}$$

$$\mathcal{T}_1(z) = -\frac{2z^2 K_0(z) + \left(z^2 + 4\right) z K_1(z) - 4}{2\pi z^4}, \tag{B.29}$$

$$\mathcal{T}_2(z) = \frac{4z^2 K_0(z) + \left(z^2 + 8\right) z K_1(z) - 8}{\pi z^6}, \tag{B.30}$$

$$\mathcal{T}_3(z) = \frac{z K_1(z) - 1}{2\pi z^2}. \tag{B.31}$$

To avoid cancellation errors, the above expressions must be evaluated using power series for small values of the argument z, see appendix D.

Appendix C Higher-order moments of the modified biharmonic Green's function

As observed in section 3.2.1, a charge can be represented perfectly as a 0-term expansion about its own center because $G(\boldsymbol{x}, \boldsymbol{y}) = -Q_0(r)/(2\pi\alpha^2)$, with $r = |\boldsymbol{x} - \boldsymbol{y}|$. Here we derive the expressions for the directional derivatives of G up to third order in the multipole basis, $\{Q_n, K_n\}$, which are necessary to form octopole moments. As in appendix B, we make use of the standard suffix notation and set $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}$.

Some preliminary expressions for the spatial derivatives of G in terms of its derivatives as a function of the radius r are given in (B.17)–(B.19). We then must obtain expressions for the derivatives of G as a function of r in the multipole basis. From the recurrence and derivative relations for modified Bessel functions [51, §10.29], we obtain

$$G'(r) = -\frac{Q_1(r)}{2\pi\alpha} , \tag{C.1}$$

$$\frac{G'(r)}{r} = \frac{Q_2(r) - K_0(\alpha r)}{4\pi} , \tag{C.2}$$

$$G''(r) - \frac{G'(r)}{r} = -\frac{Q_2(r)}{2\pi} , \tag{C.3}$$

$$\frac{G''(r)}{r} - \frac{G'(r)}{r^2} = -\alpha\frac{Q_3(r) - K_1(\alpha r)}{8\pi} , \tag{C.4}$$

$$G'''(r) - 3\frac{G''(r)}{r} + 3\frac{G'(r)}{r^2} = \alpha\frac{Q_3(r)}{2\pi} . \tag{C.5}$$

To derive the separation of variables form of the directional derivatives, we identify the point $\boldsymbol{r}$ with $re^{i\theta}$. In the following, we switch freely between identifying vectors by their suffix and complex valued notations. Let $\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}$ be vectors and λ, μ, ν denote their complex valued counterpart. Then,

$$\frac{r_i\lambda_i}{r} = \frac{e^{i\theta}\bar{\lambda} + e^{-i\theta}\lambda}{2} , \tag{C.6}$$

so that

$$\partial_\lambda G = -\frac{Q_1(r)}{4\pi\alpha}\left(e^{i\theta}\bar{\lambda} + e^{-i\theta}\lambda\right) . \tag{C.7}$$

Similarly, we have that

$$\frac{r_ir_j\lambda_i\mu_j}{r^2} = \frac{e^{i2\theta}\bar{\lambda}\bar{\mu} + e^{-i2\theta}\lambda\mu + \lambda\bar{\mu} + \bar{\lambda}\mu}{4} \tag{C.8}$$

$$\delta_{ij}\lambda_i\mu_j = \frac{\lambda\bar{\mu} + \bar{\lambda}\mu}{2} , \tag{C.9}$$

so that

$$\partial_\lambda\partial_\mu G = -\frac{Q_2(r)}{8\pi}\left(e^{i2\theta}\bar{\lambda}\bar{\mu} + e^{-i2\theta}\lambda\mu\right) - \frac{K_0(\alpha r)}{8\pi}\left(\lambda\bar{\mu} + \bar{\lambda}\mu\right) . \tag{C.10}$$

Finally, we have that

$$\frac{r_ir_jr_k\lambda_i\mu_j\nu_k}{r^3} = \frac{e^{i3\theta}\bar{\lambda}\bar{\mu}\bar{\nu} + e^{-i3\theta}\lambda\mu\nu + e^{i\theta}(\bar{\lambda}\bar{\mu}\nu + \bar{\lambda}\mu\bar{\nu} + \lambda\bar{\mu}\bar{\nu}) + e^{-i\theta}(\lambda\mu\bar{\nu} + \lambda\bar{\mu}\nu + \bar{\lambda}\mu\nu)}{8} \tag{C.11}$$

$$\frac{\delta_{jk}r_i + \delta_{ki}r_j + \delta_{ij}r_k}{r}\lambda_i\mu_j\nu_k = \frac{e^{i\theta}(\bar{\lambda}\bar{\mu}\nu + \bar{\lambda}\mu\bar{\nu} + \lambda\bar{\mu}\bar{\nu}) + e^{-i\theta}(\lambda\mu\bar{\nu} + \lambda\bar{\mu}\nu + \bar{\lambda}\mu\nu)}{2} , \tag{C.12}$$

so that

$$\partial_\lambda\partial_\mu\partial_\nu G = \frac{\alpha Q_3(r)}{16\pi}\left(e^{i3\theta}\bar{\lambda}\bar{\mu}\bar{\nu} + e^{-i3\theta}\lambda\mu\nu\right) + \frac{\alpha K_1(\alpha r)}{16\pi}\left(e^{i\theta}(\bar{\lambda}\bar{\mu}\nu + \bar{\lambda}\mu\bar{\nu} + \lambda\bar{\mu}\bar{\nu}) + e^{-i\theta}(\lambda\mu\bar{\nu} + \lambda\bar{\mu}\nu + \bar{\lambda}\mu\nu)\right) . \tag{C.13}$$

Appendix D Power series

In the closed-form expressions for the stokeslet and the stresslet, (B.24) and (B.26), the expressions for the functions $\mathcal{S}_i(z)$ and $\mathcal{T}_i(z)$, (B.27)–(B.31), are prone to cancellation errors for small arguments. To get around this, we form their power series, by combining the coefficients of the series expansions of the modified Bessel functions. The fundamental expansions needed are [51, §10.25 and §10.31]

$$I_0(z) = \sum_{n=0}^{\infty} \frac{\left(\frac{1}{4}z^2\right)^n}{(n!)^2}, \tag{D.1}$$

$$I_1(z) = \frac{z}{2} \sum_{n=0}^{\infty} \frac{\left(\frac{1}{4}z^2\right)^n}{n!(n+1)!}, \tag{D.2}$$

$$K_0(z) = -\log(z/2) I_0(z) + \sum_{n=0}^{\infty} \psi(n+1) \frac{\left(\frac{1}{4}z^2\right)^n}{(n!)^2}, \tag{D.3}$$

$$K_1(z) = \frac{1}{z} + \log(z/2) I_1(z) - \frac{z}{4} \sum_{n=0}^{\infty} (\psi(n+1) + \psi(n+2)) \frac{\left(\frac{1}{4}z^2\right)^n}{n!(n+1)!}, \tag{D.4}$$

where ψ is the digamma function, which can be evaluated recursively starting from the Euler-Mascheroni constant γ,

$$\psi(1) = -\gamma \tag{D.5}$$

$$\psi(n+1) = \psi(n) + 1/n. \tag{D.6}$$

We write the above power series on the following compact form:

$$I_0(z) = \sum_{n=0}^{\infty} i_0(n) z^{2n}, \qquad I_1(z) = \sum_{n=0}^{\infty} i_1(n) z^{2n+1}, \tag{D.7}$$

$$K_0(z) = \sum_{n=0}^{\infty} k_0(z,n) z^{2n}, \qquad K_1(z) = \frac{1}{z} + \sum_{n=0}^{\infty} k_1(z,n) z^{2n+1}, \tag{D.8}$$

where

$$i_0(n) = \frac{1}{4^n (n!)^2}, \tag{D.9}$$

$$i_1(n) = \frac{1}{2} \frac{1}{4^n n!(n+1)!}, \tag{D.10}$$

$$k_0(z,n) = \big(\psi(n+1) - \log(z/2)\big) \frac{1}{4^n (n!)^2} \tag{D.11}$$

$$k_1(z,n) = \big(2\log(z/2) - \psi(n+1) - \psi(n+2)\big) \frac{1}{4^{n+1} n!(n+1)!} \tag{D.12}$$

In our application we only have use for the functions $\mathcal{T}_i(z)$. These can now be computed for small z using

$$\mathcal{T}_1(z) = -\frac{1}{2\pi} \sum_{n=1}^{\infty} \big(2k_0(z,n) + 4k_1(z,n) + k_1(z,n-1)\big) z^{2n-2}, \tag{D.13}$$

$$\mathcal{T}_2(z) = -\frac{1}{\pi z^4} + \frac{1}{8\pi z^2} + \frac{1}{\pi} \sum_{n=2}^{\infty} \big(4k_0(z,n) + 8k_1(z,n) + k_1(z,n-1)\big) z^{2n-4}, \tag{D.14}$$

$$\mathcal{T}_3(z), = \frac{1}{2\pi} \sum_{n=0}^{\infty} k_1(z,n) z^{2n}. \tag{D.15}$$

In our code, we have empirically determined that we need to switch to power series evaluation for $\mathcal{T}_i(z)$ when $z \leq 1.5$, and that it is sufficient to truncate the sums to 11 terms. For the derivatives $\mathcal{T}_i'(z)$, we evaluate the derivatives of the above series when $z \leq 2$, with the sums truncated to 13 terms.

For the kernel-split quadrature described in section 4, we also need power series expansions of the smooth functions $\mathcal{T}_i^S$ and $\mathcal{T}_i^L$, as defined in section 4.1. These are straightforward to break out from the above expansions, by splitting the $\log(z)$ factor from the coefficients,

$$k_i(z,n) = k_i^S(n) + k_i^L(n)\log z. \tag{D.16}$$

References

[1] D. J. Acheson. *Elementary Fluid Dynamics.* Oxford University Press, New York, 1990. ISBN 9780198596790.

[2] L. af Klinteberg and T. Askham. INSE FIEM 2D, 2019. URL `https://github.com/ludvigak/inse-fiem-2d`.

[3] L. af Klinteberg and A. H. Barnett. Accurate quadrature of nearly singular line integrals in two and three dimensions by singularity swapping. *arXiv:1910.09899 [math.NA]*, 2019.

[4] L. af Klinteberg and A.-K. Tornberg. Error estimation for quadrature by expansion in layer potential evaluation. *Adv. Comput. Math.*, 43(1):195–234, 2017. doi: 10.1007/s10444-016-9484-x.

[5] L. af Klinteberg and A.-K. Tornberg. Adaptive Quadrature by Expansion for Layer Potential Evaluation in Two Dimensions. *SIAM J. Sci. Comput.*, 40(3):A1225–A1249, 2018. doi: 10.1137/17M1121615.

[6] L. af Klinteberg, F. Fryklund, and A.-K. Tornberg. An adaptive kernel-split quadrature method for parameter-dependent layer potentials. *arXiv:1906.07713 [math.NA]*, 2019.

[7] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-Explicit Methods for Time-Dependent Partial Differential Equations. *SIAM J. Numer. Anal.*, 32(3):797–823, 1995. doi: 10.1137/0732037.

[8] T. Askham. *Integral-equation methods for inhomogeneous elliptic partial differential equations in complex geometry.* PhD thesis, New York University, 2016.

[9] T. Askham. A Stabilized Separation of Variables Method for the Modified Biharmonic Equation. *J. Sci. Comput.*, 76(3):1674–1697, 2018. doi: 10.1007/s10915-018-0679-9.

[10] T. Askham and A. Cerfon. An adaptive fast multipole accelerated Poisson solver for complex geometries. *J. Comput. Phys.*, 344:1–22, 2017. doi: 10.1016/j.jcp.2017.04.063.

[11] A. H. Barnett. Evaluation of Layer Potentials Close to the Boundary for Laplace and Helmholtz Problems on Analytic Planar Domains. *SIAM J. Sci. Comput.*, 36(2):A427–A451, 2014. doi: 10.1137/120900253.

[12] A. H. Barnett, J. Magland, and L. af Klinteberg. A Parallel Nonuniform Fast Fourier Transform Library Based on an Exponential of Semicircle" Kernel. *SIAM J. Sci. Comput.*, 41(5):C479–C504, 2019. doi: 10.1137/18M120885X.

[13] J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange Interpolation. *SIAM Rev.*, 46(3):501–517, 2004. doi: 10.1137/S0036144502417715.

[14] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.*, 59(1):65–98, 2017. doi: 10.1137/141000671.

[15] G. Biros, L. Ying, and D. Zorin. The embedded boundary integral method for the unsteady incompressible Navier–Stokes equations. Technical Report TR2003-838, Courant Institute, New York University, 2002.

[16] A. Björck and V. Pereyra. Solution of Vandermonde Systems of Equations. *Math. Comput.*, 24(112): 893, 1970. doi: 10.2307/2004623.

[17] H. C. Brinkman. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Appl. Sci. Res.*, A1:27, 1947. doi: 10.1007/BF02120313.

[18] M. Causley, A. Christlieb, B. Ong, and L. Van Groningen. Method of lines transpose: An implicit solution to the wave equation. *Math. Comput.*, 83(290):2763–2786, 2014. doi: 10.1090/S0025-5718-2014-02834-2.

[19] R. Chapko and R. Kress. Rothe's Method for the Heat Equation and Boundary Integral Equations. *J. Integr. Equations Appl.*, 9(1):47–69, 1997. doi: 10.1216/jiea/1181075987.

[20] H. Cheng, Z. Gimbutas, P. G. Martinsson, and V. Rokhlin. On the Compression of Low Rank Matrices. *SIAM J. Sci. Comput.*, 26(4):1389–1404, 2005. doi: 10.1137/030602678.

[21] H. Cheng, J. Huang, and T. J. Leiterman. An adaptive fast solver for the modified Helmholtz equation in two dimensions. *J. Comput. Phys.*, 211(2):616–637, 2006. doi: 10.1016/j.jcp.2005.06.006.

[22] A. Dutt, L. Greengard, and V. Rokhlin. Spectral Deferred Correction Methods for Ordinary Differential Equations. *Bit Numer. Math.*, 40(2):241–266, 2000. doi: 10.1023/A:1022338906936.

[23] B. Fornberg, E. Larsson, and N. Flyer. Stable Computations with Gaussian Radial Basis Functions. *SIAM J. Sci. Comput.*, 33(2):869–892, 2011. doi: 10.1137/09076756X.

[24] W. L. Frank. Computing Eigenvalues of Complex Matrices by Determinant Evaluation and by Methods of Danilewski and Wielandt. *J. Soc. Ind. Appl. Math.*, 6(4):378–392, 1958. doi: 10.1137/0106026.

[25] F. Fryklund and E. Lehto. PUXdemo, 2018. URL `https://github.com/fryklund/PUXdemo`.

[26] F. Fryklund, E. Lehto, and A.-K. Tornberg. Partition of unity extension of functions on complex domains. *J. Comput. Phys.*, 375:57–79, 2018. doi: 10.1016/j.jcp.2018.08.012.

[27] F. Fryklund, M. C. A. Kropinski, and A.-K. Tornberg. An integral equation based numerical method for the forced heat equation on complex domains. *arXiv:1907.08537 [math.NA]*, 2019.

[28] A. Gillman, P. M. Young, and P.-G. Martinsson. A direct solver with O(N) complexity for integral equations on one-dimensional domains. *Front. Math. China*, 7(2):217–247, 2012. doi: 10.1007/s11464-012-0188-3.

[29] Z. Gimbutas and L. Greengard. FMMLIB2D, 2017. URL `https://github.com/zgimbutas/fmmlib2d`.

[30] L. Greengard and S. Jiang. A new mixed potential representation for the equations of unsteady, incompressible flow. *arXiv:1809.08442 [math.NA]*, 2018.

[31] L. Greengard and M. C. Kropinski. An Integral Equation Approach to the Incompressible Navier–Stokes Equations in Two Dimensions. *SIAM J. Sci. Comput.*, 20(1):318–336, 1998. doi: 10.1137/S1064827597317648.

[32] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987. doi: 10.1016/0021-9991(87)90140-9.

[33] L. Greengard and V. Rokhlin. A new version of the Fast Multipole Method for the Laplace equation in three dimensions. *Acta Numer.*, 6:229, 1997. doi: 10.1017/S0962492900002725.

[34] L. Greengard, S. Jiang, and J. Wang. On the accurate evaluation of unsteady Stokes layer potentials in moving two-dimensional geometries. 10010:1–17, 2018.

[35] S. Hao, A. H. Barnett, P. G. Martinsson, and P. Young. High-order accurate methods for Nyström discretization of integral equations on smooth curves in the plane. *Adv. Comput. Math.*, 40(1):245–272, 2014. doi: 10.1007/s10444-013-9306-3.

[36] J. Helsing. Integral equation methods for elliptic problems with boundary conditions of mixed type. *J. Comput. Phys.*, 228(23):8892–8907, 2009. doi: 10.1016/j.jcp.2009.09.004.

[37] J. Helsing and A. Holst. Variants of an explicit kernel-split panel-based Nyström discretization scheme for Helmholtz boundary value problems. *Adv. Comput. Math.*, 41(3):691–708, 2015. doi: 10.1007/s10444-014-9383-y.

[38] J. Helsing and S. Jiang. On Integral Equation Methods for the First Dirichlet Problem of the Biharmonic and Modified Biharmonic Equations in NonSmooth Domains. *SIAM J. Sci. Comput.*, 40(4):A2609–A2630, 2018. doi: 10.1137/17M1162238.

[39] J. Helsing and R. Ojala. On the evaluation of layer potentials close to their sources. *J. Comput. Phys.*, 227(5):2899–2921, 2008. doi: 10.1016/j.jcp.2007.11.024.

[40] J. Helsing and R. Ojala. Corner singularities for elliptic problems: Integral equations, graded meshes, quadrature, and compressed inverse preconditioning. *J. Comput. Phys.*, 227(20):8820–8840, 2008. doi: 10.1016/j.jcp.2008.06.022.

[41] K. Ho, S. Olver, T. Kelman, and R. M. Slevinsky. LowRankApprox.jl, 2018. URL `https://zenodo.org/record/1481098`.

[42] S. Jiang, M. C. A. Kropinski, and B. D. Quaife. Second kind integral equation formulation for the modified biharmonic equation and its applications. *J. Comput. Phys.*, 249:113–126, 2013. doi: 10.1016/j.jcp.2013.04.034.

[43] S. Kim and S. J. Karrila. *Microhydrodynamics : principles and selected applications.* Butterworth-Heinemann series in chemical engineering. Butterworth-Heinemann, Boston, 1991. ISBN 0750691735.

[44] A. Klöckner, A. Barnett, L. Greengard, and M. O'Neil. Quadrature by expansion: A new method for the evaluation of layer potentials. *J. Comput. Phys.*, 2013. doi: 10.1016/j.jcp.2013.06.027.

[45] M. C. A. Kropinski and B. D. Quaife. Fast integral equation methods for Rothe's method applied to the isotropic heat equation. *Comput. Math. with Appl.*, 61(9):2436–2446, 2011. doi: 10.1016/j.camwa.2011.02.024.

[46] E. Larsson. RBF-QR, 2015. URL `http://www.it.uu.se/research/scientific_computing/software/rbf_qr`.

[47] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proc. Natl. Acad. Sci.*, 104(51):20167–20172, 2007. doi: 10.1073/pnas.0709640104.

[48] G. R. Marple, A. Barnett, A. Gillman, and S. Veerapaneni. A Fast Algorithm for Simulating Multiphase Flows Through Periodic Geometries of Arbitrary Shape. *SIAM J. Sci. Comput.*, 38(5):B740–B772, 2016. doi: 10.1137/15M1043066.

[49] P. Martinsson and V. Rokhlin. A fast direct solver for boundary integral equations in two dimensions. *J. Comput. Phys.*, 205(1):1–23, 2005. doi: 10.1016/j.jcp.2004.10.033.

[50] M. L. Minion. Semi-implicit spectral deferred correction methods for ordinary differential equations. *Commun. Math. Sci.*, 1(3):471–500, 2003. doi: 10.4310/CMS.2003.v1.n3.a6.

[51] NIST. Digital Library of Mathematical Functions. Release 1.0.16, 2017. URL `http://dlmf.nist.gov/`.

[52] R. Ojala and A.-K. Tornberg. An accurate integral equation method for simulating multi-phase Stokes flow. *J. Comput. Phys.*, 298:145–160, 2015. doi: 10.1016/j.jcp.2015.06.002.

[53] C. S. Peskin. The immersed boundary method. *Acta Numer.*, 11:479–517, 2002. doi: 10.1017/S0962492902000077.

[54] C. Pozrikidis. A singularity method for unsteady linearized flow. *Phys. Fluids A Fluid Dyn.*, 1(9):1508–1520, 1989. doi: 10.1063/1.857329.

[55] C. Pozrikidis. *Boundary Integral and Singularity Methods for Linearized Viscous Flow.* Cambridge University Press, Cambridge, 1992. ISBN 9780511624124. doi: 10.1017/CBO9780511624124.

[56] M. Rachh and K. Serkh. On the solution of Stokes equation on regions with corners. *arXiv:1711.04072 [math-ph]*, 2017.

[57] Y. Saad and M. H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986. doi: 10.1137/0907058.

[58] K. Serkh and V. Rokhlin. On the solution of elliptic partial differential equations on regions with corners. *J. Comput. Phys.*, 305:150–171, 2016. doi: 10.1016/j.jcp.2015.10.024.

[59] D. B. Stein, R. D. Guy, and B. Thomases. Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains. *J. Comput. Phys.*, 335:155–178, 2017. doi: 10.1016/j.jcp.2017.01.010.

[60] A. Tafuni, J. Domínguez, R. Vacondio, and A. Crespo. A versatile algorithm for the treatment of open boundary conditions in Smoothed particle hydrodynamics GPU models. *Comput. Methods Appl. Mech. Eng.*, 342:604–624, 2018. doi: 10.1016/j.cma.2018.08.004.

[61] M. Wala and A. Klöckner. A fast algorithm for Quadrature by Expansion in three dimensions. *J. Comput. Phys.*, 388:655–689, 2019. doi: 10.1016/j.jcp.2019.03.024.

[62] M. Wala and A. Klöckner. Optimization of fast algorithms for global Quadrature by Expansion using target-specific expansions. *J. Comput. Phys.*, 403:108976, 2020. doi: 10.1016/j.jcp.2019.108976.

[63] J. Wang and L. Greengard. Hybrid asymptotic/numerical methods for the evaluation of layer heat potentials in two dimensions. *Adv. Comput. Math.*, 45(2):847–867, 2019. doi: 10.1007/s10444-018-9641-5.

[64] B. Wu, H. Zhu, A. Barnett, and S. Veerapaneni. Solution of Stokes flow in complex nonsmooth 2D geometries via a linear-scaling high-order adaptive integral equation scheme. *arXiv:1909.00049 [math.NA]*, 2019.

[65] Y. Zhang and A. Gillman. A fast direct solver for boundary value problems on locally perturbed geometries. *J. Comput. Phys.*, 356:356–371, 2018. doi: 10.1016/j.jcp.2017.12.013.