HIGH-ORDER HYBRID METHODS USING GREEN'S FUNCTIONS AND FINITE ELEMENTS

BY

NATALIE NICOLE BEAMS

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Theoretical and Applied Mechanics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Paul Fischer, Chair
Professor Luke Olson, Director of Research
Assistant Professor Andreas Klöckner
Associate Professor Matthew West

Abstract

Green's functions provide an elegant mathematical framework for linear partial differential boundary value problems by directly defining solutions in integral form. For a Green's function that satisfies the desired boundary conditions, this form is a convolution of the Green's function with the right hand side. However, straightforward numerical computation of the convolution generally results in a prohibitively costly algorithm. This dissertation focuses on combining Green's function–based computations with finite element (FE) methods to alleviate their computational challenges while enhancing the representation capability of standard FE solutions on structured or semi-structured meshes. Two main areas of application are considered: a new finite-element-based method for N-body calculations and a new high-order method for elliptic interface problems on non-conforming volume meshes.

First, the FE-based particle-particle–particle-mesh (FE-P^{3}M) method for N-body problems is introduced. P^{3}M methods are designed to compute quantities (e.g. potentials or forces) that are sums of discrete Green's functions and represent the effects of N interacting bodies. To avoid an N^2 operation when the quantity is desired at all body locations (as is often the case), P^{3}M methods decompose the potential into short-range interactions between near neighbors and long-range interactions that are smooth and readily solved by mesh-based numerical methods. Instead of taking the traditional "Ewald-based" approach of using Gaussian screen functions to accomplish this decomposition, the FE-P^{3}M method builds specially designed polynomial screens on an introduced finite element mesh. Due to this form of the screens, the long-range component of the potential is accurately resolved with a finite element method. When compared to classic P^{3}M methods, which rely on the fast Fourier transform (FFT), the FE-P^{3}M method allows for more flexible boundary conditions and possibly much lower communication costs in a parallel implementation. The method is described in detail, its cost and memory requirements are discussed, and its accuracy is demonstrated.

The remaining chapters consider Green's functions in the context of layer potential solutions to homogeneous boundary value problems and the resulting integral equations (IEs) formulated on the boundary of the domain. A high-order coupling of finite elements and integral equations is presented for the solution of elliptic interface problems on embedded domains. This FE-IE method requires no special basis functions

nor modifications to handle homogeneous or non-homogeneous jump conditions, and retains high-order convergence close to the embedded interface. The interface or embedded boundary conditions are enforced directly at the discretization points of the embedded surface mesh. High-order numerical convergence is demonstrated for both interior and exterior embedded domain problems, with a novel splitting defined and analyzed for the latter. Theoretical error convergence, existence and uniqueness of the decomposition, and numerical solution considerations are discussed. The new FE-IE interface method is then shown to be built from interior and exterior FE-IE subproblems. Finally, an extension to Stokes flow around embedded objects in two and three dimensions is presented.

Acknowledgements

Many people have contributed to the development of this dissertation, directly and indirectly. Of course, I am very thankful to all of my committee members, especially my advisors Profs. Luke Olson and Andreas Klöckner, without whom I would not have made it to this point. I am also indebted to the many outstanding teachers I have had throughout the years, who always encouraged me in my academic pursuits. My decision to attend graduate school was greatly influenced by Prof. David Miller of the University of Oklahoma, who served as a wonderful mentor during my undergraduate career.

I am grateful for the colleagues I have had throughout graduate school. My humble thanks are especially due to Dr. Jessie Bock Lofton, Dr. Samantha Knoll, and Dr. David Buchta for their friendship and for always being willing to lend an ear when things weren't going well. I am also thankful for all the useful mathematical discussions with Dr. Kaushik Kalyanaraman and Matt Wala.

Of course, I am forever thankful for the love and support of my parents, David and Roberta, and my sisters, Deborah and Alli.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Green's functions provide an elegant mathematical framework for solutions to linear differential equations. Through Green's functions, the influence of boundary conditions and the right hand side on an equation's solution is clearly defined.

For a linear partial differential operator $\mathcal{L}$, the Green's function solves

$$\mathcal{L}G(x, x_0) = \delta(x - x_0) \tag{1.1}$$

where $\delta(x-x_0)$ is the Dirac delta function that is only non-zero at the point x_0. That is, $G(x, x_0)$ represents the influence of a point force located at x_0 on the solution at x.

Green's functions allow us to directly write down the solution to a boundary value problem (BVP) in an integral form. For a Green's function that satisfies the boundary conditions, this form is simply a convolution of the Green's function and the right hand side — the limit of "summing up" the effects of the point forces throughout the whole domain. As a result, Green's function solutions are global by nature: the aforementioned convolution must be performed at every point where we wish to obtain a numerical solution. This can lead to prohibitively costly algorithms for computation. In addition, not all differential operators permit a convenient closed-form Green's function.

In the case of homogeneous BVPs, where the right hand side is zero, we can avoid the need for convolution in the entire domain. We craft the solution in terms of layer potentials, in which an unknown density function, which exists only on the boundary of the domain, is integrated against a kernel derived from the free-space Green's function (that is, with the only boundary conditions being on the behavior as $x \to \infty$). Using the boundary conditions of the problem and the limiting behavior of the layer potential representation from each side of the boundary, we write an integral equation for the unknown density function, which in turn defines the solution to the original BVP. Such integral equation (IE) formulations assume arbitrary smooth geometries and require careful treatment in the presence of corners [1]. In a numerical implementation, this could take the form of extensive mesh refinement near the corners. From a practical standpoint, these methods work

best for homogeneous equations. Indeed, with a non-zero forcing function, such IE formulations once again face the computational hurdle of volume integration for every evaluation point.

In many ways, finite element methods stand in contrast to Green's function–based methods. Green's functions are global and matrices from layer potential IE discretizations are dense; finite element representations are local and the matrices are sparse. Layer potential IE methods are equally suited to interior or exterior problems; finite element methods are poorly suited to exterior problems due precisely to the local basis functions that are an advantage when creating a sparse system to solve. Layer potential IE methods are built on the assumption of smooth domains; finite elements work well in convex polygonal domains, but require mesh refinement and/or the use of modified elements to represent curved boundaries [2, 3].

It is the opposite nature of these methods which leads us to ask whether they can be combined in such a way as to expose the best features of both for the problem at hand. We consider two major areas of applications: N-body problems and embedded boundary and interface problems.

1.1 N-body problems

N-body interactions arise in a range of applications, including electrostatics, molecular dynamics, and plasma dynamics: systems of bodies where the solution of some quantity of interest at the location of one body depends on the location of all the other bodies. In this case, the Green's function solution reduces to a discrete sum over all bodies. For example, consider the electrostatics problem of finding the electrostatic potential Φ of a collection of N point charges with strengths Q_j. The governing equation is

$$-\nabla \cdot \nabla \Phi = \sum_{j=1}^{N} Q_j \delta(\mathbf{x}_j), \tag{1.2}$$

where $\delta(\mathbf{x}_j)$ is the Dirac delta function. For free-space boundary conditions in three dimensions, the Green's function is

$$G(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{4\pi|\mathbf{x}_i - \mathbf{x}_j|} \tag{1.3}$$

and the total electrostatic potential at each particle location is

$$\Phi(\mathbf{x}_i) = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \frac{Q_j}{4\pi|\mathbf{x}_i - \mathbf{x}_j|}. \tag{1.4}$$

If we wish to find the potential at each charge location (e.g., to calculate the total potential energy), a direct sum would incur N^2 operations at best, with additional "image" charges possibly needed to satisfy

the boundary conditions. Worse still, in the case of periodic boundary conditions, the sum becomes conditionally convergent. Not surprisingly, then, there has been great interest in the development of more efficient algorithms for computing sums such as (1.4).

The most widely used methods for N-body problems are generally classified as either tree-based, such as the fast multipole method (FMM) [4], or mesh-based (sometimes called "particle-in-cell"), such as the particle-particle–particle-mesh (P^3M) method [5], its popular variant, the particle-mesh-Ewald (PME) method [6, 7], and other "Ewald-like" operator splitting methods such as the General Geometry Ewald-like Method (GGEM) [32, 33] and the Multilevel Summation Method (MSM) [34, 35, 36, 37].

In the FMM, particles are grouped within multipole expansions to provide an accurate representation of their combined influence at a distance, thus limiting the number of terms needed to explicitly compute the interactions. The resulting algorithm scales with $\mathcal{O}(N)$ complexity, although the coefficient in this scaling can be large, especially if a high-order multipole expansion is required for the algorithm [8]. The method also extends to systems with more complicated kernels, such as Stokes flow [9, 10, 11].

Mesh-based methods take an N-body system which exists in continuous space and places it on an introduced discrete mesh. Long-range interactions between particles in the system are then coupled through the mesh instead of explicitly pairwise, and we can use standard mesh-based numerical methods to solve the governing equations of the system. However, we note that the Green's function (1.3) is singular when $\mathbf{x}_i = \mathbf{x}_j$, and such standard mesh-based methods will necessarily smooth out this singular behavior. P^3M methods correct for this in the "particle-particle" part of the computation, which is only needed in a small area around each particle. This is described in more detail in Section 2.2.

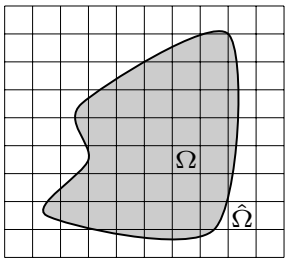
In previous work by Beams, Olson, and Freund, a finite-element-based P^3M method was introduced [12]. This method uses specially-designed polynomial functions to interface between the particle locations and the mesh. It combines Green's functions and finite elements by building these assignment functions so that they are exactly resolved on the finite element mesh while still reproducing the long-range Green's function behavior of the potential due to a particle, to the order allowed by the degree of the polynomial. The method is further explained in Sections 3.1–3.2 and demonstrated in Section 3.3.

1.2 Embedded domains and structured meshes

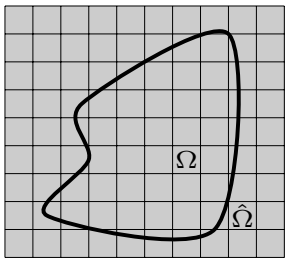
One of the benefits of finite element methods is their flexibility with regards to domain geometry. However, creating a satisfactorily-conforming mesh for many domains is not a straightforward task; this is especially a concern in time-dependent problems if re-meshing is needed at every or many time steps. Furthermore, there

are practical computational advantages of being able to use a structured mesh. The resulting matrices have a regular sparsity pattern, making preconditioning and solving more efficient compared to fully unstructured problems. It is also a trivial task to place points within the mesh, simplifying the process of evaluating the solution at non-mesh points.

Due to these advantages, there has been much interest in embedded domain finite element methods, where the problem domain Ω is placed into a larger, rectangular domain $\hat{\Omega}$ (see Figure 1.1) and the problem is modified to be solved on the new domain while still satisfying (to some extent) the boundary conditions on the original boundary $\partial\Omega$. Examples include finite cell methods [13, 14], which integrate the terms of the functional to be minimized only on the original domain Ω, requiring careful treatment of elements which are only partially inside Ω, especially those containing only small pieces of Ω. The boundary conditions are enforced weakly via the introduction of Lagrange multipliers in the functional or similar methods. In order to alleviate the need for this careful treatment of some elements, fictitious domain methods integrate outside Ω as well, extending the right hand side as necessary [15, 16]. To enforce essential boundary conditions, such methods again use Lagrange multipliers or penalty methods.



(a) In finite cell methods, the integration of the weak equations conforms to the domain, even though the mesh is nonconforming.



(b) In fictitious domain methods, integration of the weak equations occurs on the entire fictitious domain $\hat{\Omega}$, with the right hand side extended as necessary.

Figure 1.1: Problem domain Ω embedded in a rectangular grid on domain $\hat{\Omega}$ for finite cell and fictitious domain methods

Previous work combining finite element methods with integral equations for irregular domains was done by Biros et al. [17], who solved the Stokes equations on a regular mesh using a method based on work by Mayo [18] for the Poisson and biharmonic equations. The authors find a layer-potential representation that exists in both the actual domain Ω and $\hat{\Omega}\backslash\Omega$, but is discontinuous across the domain boundary $\partial\Omega$. Using known information about the jumps in this representation and its derivatives at the boundary $\partial\Omega$, they calculate modifications to the stencil for the differential operator of the PDE so that they can solve for this integral representation on the volume mesh.

Rüberg and Cirak, in contrast, separate the solution into two parts: a finite element solution found on

the regular mesh, and an integral equation solution defined by the boundary of the actual domain [19]. It is worth noting that in contrast with Lagrange multiplier and similar methods, the Dirichlet boundary conditions in this method are enforced exactly at every discretization point on the embedded boundary, and no extra variables are introduced or additional terms added to the finite element functional. It is this type of decomposition that we consider in this dissertation. We show how to obtain high order coupling for interior and exterior embedded domain problems in Sections 4.2 and 4.3.

From domains with exclusions, we consider the natural extension to interface problems on non-conforming meshes. We now consider two domains Ω_i and Ω_e separated by an interface Γ, subject to jump conditions on the value of the solution and its normal derivative across the interface. This is illustrated in Figure 1.2. The coefficients of the linear differential operator and the forcing function may both be discontinuous across Γ. The model problem is

$$\begin{aligned} -\beta\nabla\cdot\nabla u(x) &= f(x) \quad \text{in} \quad \Omega^i \cup \Omega^e, \quad \text{with} \\ u^i(x) &= cu^e(x) + a(x) \quad \text{on} \quad \Gamma, \quad \text{and} \\ \frac{\partial u^i(x)}{\partial n} &= \kappa \frac{\partial u^e(x)}{\partial n} + b(x) \quad \text{on} \quad \Gamma. \end{aligned} \tag{1.5}$$

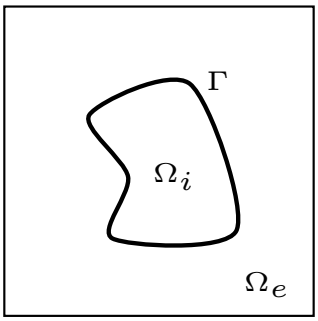
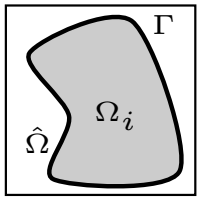


Figure 1.2: Two domains Ω_i and Ω_e separated by interface Γ.

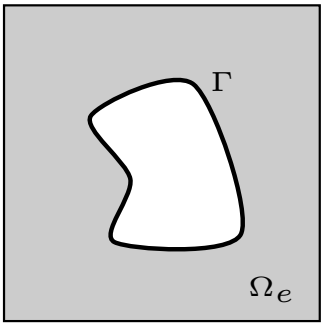
LeVeque and Li introduced the immersed interface method (IIM) to solve elliptic interface problems with discontinuous coefficients and singular sources using finite differences on a regular Cartesian grid [20]. In the IIM, finite difference stencil is modified to satisfy the jump conditions at the boundary. A summary of immersed interface (and closely-related immersed boundary) methods is given in a review paper by Mittal and Iaccarino [21].

As perhaps expected, the IIM has also been extended to finite element methods, often called "immersed finite element" methods (IFEM) [22, 23]. As the IIM changes the finite difference operator, so does the IFEM change the finite element representation — by creating special basis functions in elements crossed by

the interface, chosen to satisfy the jump conditions at the interface. The method can be modified to handle non-homogeneous jump conditions [24, 25].



(a) Interior problem on Ω_i embedded in a fictitious domain $\hat{\Omega}$.



(b) Problem in Ω_e is treated as a domain with an exclusion.

Figure 1.3: Splitting the interface domains.

In contrast, the FE-IE method herein does not require the creation of special basis functions to satisfy the boundary conditions. It also does not require the imposition of additional jump conditions to use higher order basis functions, as in [26]. Instead, we consider two separate problems: one an interior problem embedded in a rectangular fictitious domain $\hat{\Omega}$, as done in Section 4.2, the other a domain with an exclusion (with interior domain Ω_i the excluded area) as considered in Section 4.3. This is illustrated in Figure 1.3. We then proceed to apply the FE-IE method on each problem separately, coupling through the shared boundary to enforce the jump conditions.

Finally, we consider extending the FE-IE method to three-dimensional incompressible Stokes flow, which solves

$$\begin{aligned} \mu\nabla^2\mathbf{u} - \nabla p + \mathbf{f} &= 0 \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{1.6}$$

for velocity $\mathbf{u}$, viscosity μ, and pressure p in a domain Ω.

The IIM has been extended to Stokes flow [27] with rigid boundaries as well as interfaces [28]. Wagner et al. apply the eXtended finite element method (XFEM) to the problem of rigid spheres in Stokes flow [29]. Similar to IFEM, XFEM involves changing the basis in elements near the boundaries of the spheres, in this case enriching the space to better capture the near-field behavior of the fluid. The work of Zhang et al. for fluid-structure interaction is also called an immersed finite element method, but is more similar to the finite-difference-based immersed boundary method of Peskin [30], though it uses a higher-order delta

function representation to interpolate between the Lagrangian embedded domain and the Cartesian Eulerian grid [31]. We show the extension of the exterior problem to the Stokes equations in Section 6.2.

Chapter 2

Current P^{3}M methods for N-body problems

In this chapter, we examine the pieces of existing particle-particle–particle-mesh (P^{3}M) methods, in preparation for the introduction of the new FE-based P^{3}M method in Chapter 3. As mentioned in Chapter 1, mesh-based N-body methods couple long-range interactions through an introduced mesh, allowing the use of standard mesh-based numerical methods. First, we consider the simplest form of mesh-based method, which does not include any explicit pairwise calculations: the particle-mesh (PM) method (often called particle-in-cell or PIC in certain applications).

2.1 The particle-mesh method

PM solvers have three basic steps [5]:

1. Assign particles to mesh with assignment function,
2. Solve governing equation on mesh, and
3. Interpolate field variable back to particle locations.

We next consider each step in more detail for the electrostatic potential problem.

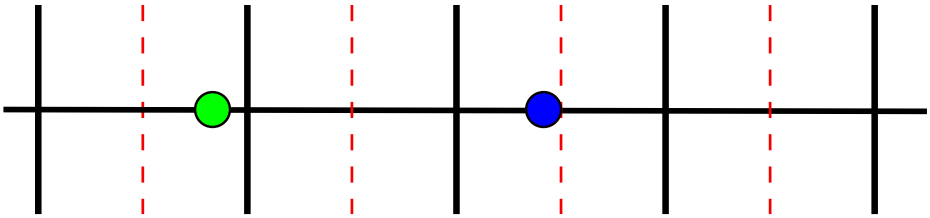
Step one, particle assignment, involves determining how the particles existing in continuous space will be distributed to the discrete locations of the introduced mesh. This is done with an assignment function such that the mesh-based charge density ρ is

$$\rho(\mathbf{x}) = \sum_{i=1}^{N} Q_i W(\mathbf{x}_i^{\mathrm{c}} - \mathbf{x}) \tag{2.1}$$

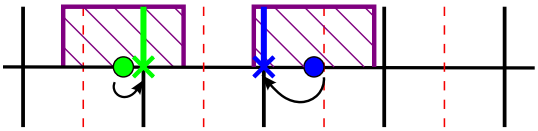
for N charges with strength Q_i. The assignment function $W(\mathbf{x}_i^{\mathrm{c}} - \mathbf{x})$ is a function of the offset of the particle location $\mathbf{x}_i^{\mathrm{c}}$ and the mesh point $\mathbf{x}$. It is responsible for distributing some fraction of the particle to the nearest mesh point(s) as determined by the value of the assignment function at each mesh point. We view

Portions of this chapter were previously published in the SIAM Journal on Scientific Computing, **38** (3) [12]

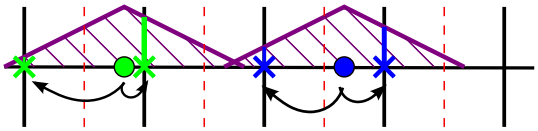
this as finding a mesh-based approximation to our continuous system; each particle is approximated by some weighted combination of particles located at the mesh points. This is shown for one dimension in Figure 2.1. In the simplest case, the entire particle is assumed to be at the mesh point nearest to it, and higher-order assignment schemes distribute particles among an increasing number of points.



(a) Two particles and an introduced mesh in one dimension. The mesh solid lines denote mesh points; dashed lines indicate the dividing line of the mesh "cells".



(b) Mesh assignment by Nearest Grid Point (NGP).



(c) Mesh assignment by Cloud-in-Cell (CIC).

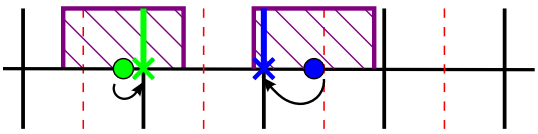
Figure 2.1: Two options for assignment of particles to an introduced discrete mesh. The value of the assignment function at each mesh point defines the fraction of the particle assigned to that point.

For step two, we must solve the governing equation for our newly mesh-based problem. In the case of electrostatic potential, that means solving

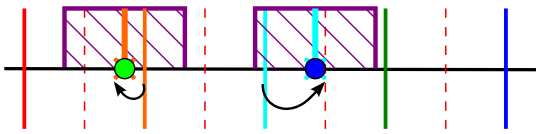
$$-\nabla^2 \Phi^{\mathrm{sm}} = \rho, \tag{2.2}$$

where here Φ^{sm} indicates that we are solving for a smooth or mesh-based potential (instead of the actual potential, which is singular if evaluated at particle locations). To solve this problem, we can utilize any numerical method in our arsenal (finite differences, finite elements, transform methods, etc.). Often, the fast Fourier transform (FFT) is employed, especially in the case of periodic boundary conditions, though this is not inherently part of the PM scheme.

Finally, in step three, we interpolate the desired field variable back to the actual locations of our charges $\mathbf{x}_i^{\mathrm{c}}$. That is, we want some weighted combination of the values at mesh points which gives an approximation to the actual value at the particle locations. Often, the potential is calculated in order to find forces on particles due to the other particles. In this case, an overwhelmingly popular choice for interpolation is to use the same assignment function as in step one, which ensures conservation of momentum. The NGP assignment-interpolation pair are illustrated in Figure 2.2.



(a) Mesh assignment by Nearest Grid Point (NGP).



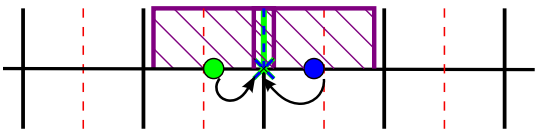
(b) Interpolation of field values to particle locations.

Figure 2.2: NGP assignment and its momentum-conserving counterpart, NGP interpolation.

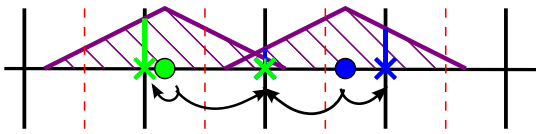
Screening effect of PM

Assigning the particles to specific mesh points projects the particle locations from a continuous space to a discrete space. This not only causes inaccuracies, but introduces an artificial screening effect by which two particles that are very close together are "screened" from each other. For example, consider two particles that are in the same cell in the case of NGP assignment. Both particles are assigned to the same grid point and are thus treated by the mesh as one particle located at the common grid point, as shown in Figure 2.3a. In continuous space, the potential and force induced at one particle location due to the other particle would increase asymptotically toward infinity as the particles move closer together. In the discrete NGP version, the potential remains the same independent of the particles' locations within their common cell; furthermore, the force between the particles is zero. In other words, two particles coupled by a mesh in this way cannot "see" each other at all: each is screened from the other.

Increasing the order of the assignment scheme helps to correct for this phenomenon, as only a fraction of each particle in the same cell is assigned to the same grid point. However, due to the fact that higher-order assignment functions have larger support and distribute particles to more mesh points, the screening effect will appear in particles with larger distances between them. For example, some screening will occur even in particles separated by a full cell, as shown in Figure 2.3b.



(a) Particles in the same cell cannot "see" each other at all with NGP assignment.



(b) Fractions of two particles are screened in CIC assignment.

Figure 2.3: Screening effect of PM results in dramatic reduction of forces between particles which are very close together.

The end result is that standard PM schemes cannot resolve small-scale features on the size of a mesh cell. In some cases, this may be acceptable. However, if we wish to compensate for this screening effect, we can employ the particle-particle–particle-mesh (P^3M) method [5], which we will now describe.

2.2 The particle-particle–particle-mesh method

The fundamental idea behind the particle-particle–particle-mesh method is to couple a particle-mesh calculation with a limited number of particle-particle interactions in order to compensate for the screening effect and lack of short-range resolution in PM. It can be thought of as explicitly "putting back" these short-range effects which are lost in the particle assignment process. Thus, the total potential at the location of particle i is now

$$\Phi_i = \Phi_i^{\text{sr}} + \Phi_i^{\text{sm}}. \tag{2.3}$$

where Φ_i^{sm} is the PM potential and Φ_i^{sr} is the short-range component, defined as

$$\Phi_i^{\text{sr}} = \sum_{\substack{\{j | R_{ij} \leq R_c\} \\ j \neq i}} \Phi_{ij}^{\text{sing}} - \Phi_{ij}^{\text{ref}} \tag{2.4}$$

for the subset of particles whose distance from particle i are within some cutoff radius R_c. Φ_{ij}^{ref} is a reference potential which approximates the potential of the PM calculation for a single particle.

This means that for particles whose distance $R_{ij} > R_c$, the PM calculation is considered sufficient and $\Phi_{ij} = \Phi_{ij}^{\text{sm}}$. When $R_{ij} < R_c$ — the region where the screening effects are large and the PM calculation is inaccurate — we are essentially replacing the contribution of that particle by the singular potential:

$$\Phi_{ij} = \Phi_{ij}^{\text{sing}} - \Phi_{ij}^{\text{ref}} + \Phi_{ij}^{\text{sm}} \approx \Phi_{ij}^{\text{sing}} - \Phi_{ij}^{\text{sm}} + \Phi_{ij}^{\text{sm}} = \Phi_{ij}^{\text{sing}} \quad \forall \{j \,|\, R_{ij} < R_c\} \tag{2.5}$$

since $\Phi_{ij}^{\text{ref}} \approx \Phi_{ij}^{\text{sm}}$. In other words, we are removing the suspect PM effects of particles near to the ith particle and replacing them with the "proper" singular effects which cannot be resolved on our discrete mesh, effectively reinstating the effects which the mesh had screened out.

P^{3}M can be viewed as belonging to a larger class of methods which rely on a splitting to more efficiently tackle N-body problems. That is, we have split the N^2 problem into two parts: the PM calculation, which involves all particles in one solution, and the short-range effects present at each particle location, which involve only small subsets of nearest neighbor particles instead of the entire collection of particles.

Another splitting: the Ewald sum

A famous splitting for electrostatic potentials and other $1/R$-like kernels is the Ewald sum [39]. Whereas in P^{3}M, we think of compensating for screening effects which arise due to the nature of the PM calculation, in the Ewald sum, we consider deliberately adding specific screening functions in order to split the calculation

into two manageable parts.

The Ewald sum introduces Gaussian screening functions, such that the density contribution of an individual particle is

$$\rho_i^{\mathrm{G}}(\mathbf{x}) = Q_i \frac{4a^2}{\sqrt{\pi}} e^{-a^2|\mathbf{x}-\mathbf{x}_i|^2}, \tag{2.6}$$

where Q_i is a constant associated with the particle in question, e.g., the strength of the charge in a group of Coulombic charges. The parameter a is chosen to balance the required terms in the short-range and smooth sums. These Gaussian screening functions are introduced into the problem everywhere a particle is located in continuous space, as illustrated in Figure 2.4.

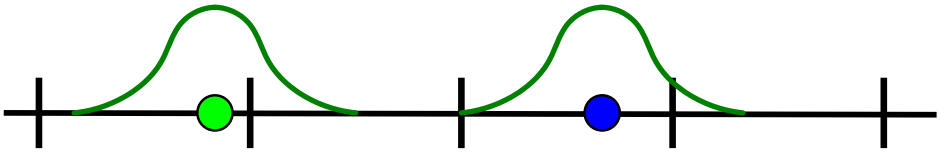


Figure 2.4: Introduction of Gaussian screening functions at particle locations to enable the Ewald splitting of the potential.

Our short-range component is now defined as the difference between the singular potential and the potential of screening function, Φ_i^{sc}:

$$\Phi_i^{\mathrm{sr,G}} = \Phi_i^{\mathrm{sing}} - \Phi_i^{\mathrm{sc,G}} \tag{2.7}$$

Such screens are chosen for two reasons: their applicability to Fourier methods for solving for the potential, and because they result in an exponentially-decaying short-range component for the particle-particle part of the calculation. This is observed by solving Poisson's equation with free-space boundary conditions (used because we are considering only one screening function, isolated in space):

$$-\frac{1}{R}\frac{d^2(R\Phi_i^{\mathrm{sc,G}})}{dR^2} = Q_i \frac{4a^2}{\sqrt{\pi}} e^{-a^2R^2} \tag{2.8}$$

$$\frac{d}{dR}(R\Phi_i^{\mathrm{sc,G}}) = Q_i \frac{2}{\sqrt{\pi}} e^{-a^2R^2} + C_1 \tag{2.9}$$

$$\Phi_i^{\mathrm{sc,G}} = Q_i \frac{\mathrm{erf}(aR)}{R} + C_1 + \frac{C_2}{R}. \tag{2.10}$$

If we choose $C_1 = C_2 = 0$, then as $R \to \infty$, $\Phi_i^{\mathrm{sc,G}} \to Q_i/R = \Phi_i^{\mathrm{sing}}$. Thus the short-range component is

$$\Phi_i^{\mathrm{sr,G}} = Q_i \frac{\mathrm{erfc}(aR)}{R}, \tag{2.11}$$

which decays rapidly as R grows larger. Again, we need only consider a small subset of particles within R_c

at each particle location $\mathbf{x}_i^{\mathrm{c}}$.

At this point, we have introduced a screening function for every particle in our system and subtracted it from the singular potential of each particle. This means we must also add each screen back into our calculation somehow in order to retain the original summation. We do this by considering the potential of all screens and, for periodic boundary conditions, all their periodic images. In other words, we must solve Poisson's equation again, this time with a right-hand side $\rho^{\mathrm{G}} = \sum_i \rho_i^{\mathrm{G}}$ including all screens, and with periodic boundary conditions. This is ideally solved by the Fourier transform:

$$\rho^{\mathrm{G}}(\mathbf{x}) = \frac{1}{L^3} \sum_{i=1}^{N} Q_i \frac{4a^2}{\sqrt{\pi}} e^{-a^2|\mathbf{x}-\mathbf{x}_i|^2} \tag{2.12}$$

$$\hat{\rho}^{\mathrm{G}} = \kappa^2 \hat{\Phi}^{\mathrm{sm,G}}, \tag{2.13}$$

so we say

$$\Phi^{\mathrm{sm,G}} = -\frac{4\pi}{L^3} \sum_{i=1}^{N} \sum_{\boldsymbol{\kappa}\neq 0} \frac{Q_i e^{-\kappa^2/4a^2}}{\kappa^2} e^{i\boldsymbol{\kappa}\cdot(\mathbf{x}-\mathbf{x}_i)}. \tag{2.14}$$

The sum in (2.14) is often referred to as the reciprocal sum, as it is completed in reciprocal space. The short-range sum in (2.11), in comparison, is referred to as the real sum.

There is one correction left to make if we want to find the potential at a particle location $\mathbf{x}_i^{\mathrm{c}}$. The potential due to the particle itself should not included, but in our screen potential solution, we have included all screens. We must then subtract the potential due to the ith screen. This is the self correction term, and it ensures that the particle cannot exert a force on itself. We find it by evaluating the screen potential $\Phi^{\mathrm{sc,G}}$ as $R \to 0$:

$$\Phi_i^{\mathrm{self,G}} = \lim_{R\to 0} Q_i \frac{\operatorname{erf}(aR)}{R} \tag{2.15}$$

$$= Q_i \frac{2a}{\sqrt{\pi}}. \tag{2.16}$$

Now we can combine the short-range, smooth, and self correction terms and find the potential at $\mathbf{x}_i^{\mathrm{c}}$:

$$\Phi(\mathbf{x}_i^{\mathrm{c}}) = \underbrace{\sum_{\substack{\{j|R_{ij}\leq R_c\}\\ j\neq i}} Q_j \operatorname{erfc}(a|\mathbf{x}_j - \mathbf{x}_i|)}_{\text{short-range}} \underbrace{- \frac{4\pi}{L^3} \sum_{j=1}^{N} \sum_{\boldsymbol{\kappa}\neq 0} \frac{Q_j e^{-\kappa^2/4a^2}}{\kappa^2} e^{i\boldsymbol{\kappa}\cdot(\mathbf{x}_i-\mathbf{x}_j)}}_{\text{smooth}} \underbrace{- Q_i \frac{2a}{\sqrt{\pi}}}_{\text{self correction}}. \tag{2.17}$$

It is important to note that the Ewald sum, while very useful for turning an infinite sum (for periodic boundary conditions) into an N^2 operation, does not involve an artificial mesh and thus cannot benefit from the many fast numerical methods available to discrete mesh-based systems.

Mesh-based Ewald: PME and SPME

The particle-mesh-Ewald (PME) method [6] can be viewed as the combination of P^3M and the Ewald sum. That is, we introduce an artificial mesh into our domain and replace each particle with a weighted combination of particles located at mesh points (assigning fractional particles to each mesh point). Each of these mesh-bound particles gets its own Gaussian screen, as shown in Figure 2.5, meaning each fractional charge resulting from mesh assignment can be viewed as both a fractional charge and a fractional screen. The calculation proceeds in the same way as for P^3M, but with the specifics of each calculation determined

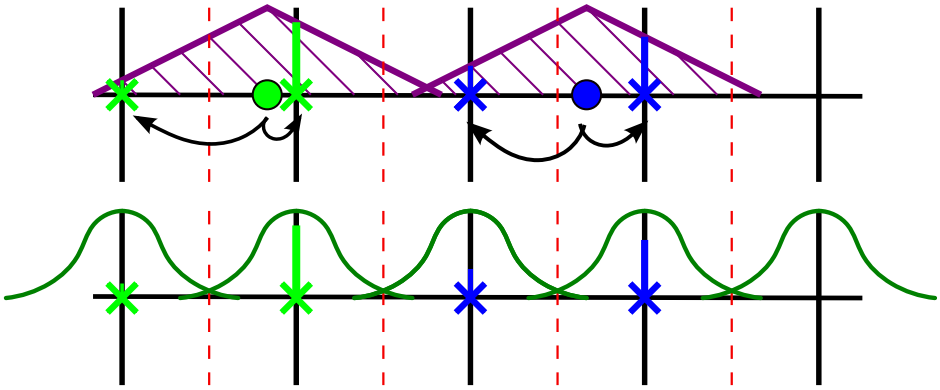


Figure 2.5: Introduction of Gaussian screening functions at each mesh point for PME. The two particles shown are assigned to the two nearest mesh points via CIC assignment, resulting in a total screen which is a linear combination of the two Gaussian screens located at the mesh points.

by the terms of the Ewald sum. The short-range effects are now calculated by the Ewald expression for the real part of the sum (2.11). Because we have introduced external screening functions, the self term given by (2.16) must be accounted for as well.

The most important shift, however, is in the smooth or mesh-based portion of the calculation, which calculates the potential of the weighted combination of Gaussian screens determined from charge assignment. As a result, the expression for the reciprocal sum (2.14) is used to calculate the smooth potential for our collection of mesh-bound Gaussian screens. Assuming we have introduced an evenly-spaced mesh, the fast Fourier transform (FFT) reduces the computational cost to $\mathcal{O}(N \log N)$.

As with P^3M, we are free to choose both our assignment function and our interpolation method for moving between continuous space and mesh space. The original PME used Lagrangian assignment and interpolation, preserving the moment-conserving constraint of Hockney and Eastwood [6], while the popular smooth PME (SPME) variant requests more smoothness and instead uses B-splines [7].

2.3 Method considerations

Any method relying on the FFT will face challenges in large-scale parallel implementation, as global communication requirements of the FFT inhibit the parallel scalability. While high-performance FFT solutions exist [40] and are effective in many settings, extreme-scale performance models suggests that global multidimensional FFTs present potential bottlenecks in scalability [41]. There are several approaches to improving Ewald-type methods to make them less restricted by the constraints of the FFT. For example, coarsening strategies reduce the number of grid points by using a staggered mesh [42]. In addition, multilevel approaches to Ewald summation [43] have increased locality in the FFT calculations while resulting in only a small increase in the total work of the FFT.

Another limitation of current evaluation methods is the application of general boundary conditions (or homogeneous boundary conditions on irregular domains). The FMM is naturally formulated for free-space boundary conditions, needing to be able to exploit something like the method of images for homogeneous Dirichlet or Neumann conditions. PME methods, which rely on the FFT, will be most natural for periodic domains, but they can be extended to free space [5, 44, 45]. However, this is done at great additional cost: doubling the size of the domain to remove periodic effects from the FFT results in doubling the work of the mesh solve just to change the boundary conditions.

The FE-P^{3}M method aims to address some of these concerns by allowing for any general boundary conditions and avoiding the need for the FFT. We introduce the method and demonstrate its application in the next chapter.

Chapter 3

FE-P^{3}M: The method of polynomial screens

The goal of the FE-based P^{3}M approach is to modify the P^{3}M method to render the mesh-based problem particularly well-suited to finite element (FE) methods, removing the dependence on the FFT. For comparison with classic P^{3}M and PME methods, we consider the sum for the electrostatic potential (1.4) of N charges with strengths Q_j, extended to periodic boundary conditions on a cube with sides of length L:

$$\Phi(\mathbf{x}_i) = \sum_{\mathbf{n}=-\boldsymbol{\infty}}^{\infty} \sum_{\substack{i,j=1 \\ i\neq j, \mathbf{n}=0}}^{N} \frac{Q_j}{|\mathbf{x}_i - \mathbf{x}_j + \mathbf{n}L|}. \tag{3.1}$$

Here $\mathbf{n}$ indicates that the sum is over all periodic images as well as the original N particles. However, the FE-based method also extends to other boundary conditions and domains, as will be discussed later and is an advantage of the method over standard P^{3}M.

3.1 Description of method

Recall from Chapter 2 that the Ewald decomposition is often viewed through the construction of a screen potential to define the corresponding short-range and long-range potentials and that the usual PME formulation uses a Gaussian screen function, yielding a short-range potential that is straightforward to compute. Furthermore, the mesh potential satisfies the Poisson problem with a right hand side built from these screens ρ_i:

$$-\nabla \cdot \nabla \Phi^{\mathrm{sm}} = \sum_i \rho_i(\mathbf{x}). \tag{3.2}$$

Equation (3.2) is then optimally solved using FFTs on a mesh.

We instead propose screening functions $\rho_i(\mathbf{x})$ that are order q piecewise polynomials on the mesh. The screens are thus represented exactly by polynomial finite element basis functions of order $\geq q$. The corresponding potential is then solved with (3.2) using a finite element method with basis functions of order $p = q + 2$.

Portions of this chapter were previously published in the SIAM Journal on Scientific Computing, **38** (3) [12]

Next, we describe the details of the method, following the four basic steps of P^{3}M methods: assignment of charges to the mesh, solving for the smooth potential on the mesh, transferring the potential back to the charge locations, and calculating the point-to-point (short-range) interactions. A high-level synopsis of the algorithm is described in Algorithm 1 to illustrate the structural pieces of our approach.

Algorithm 1: Polynomial Screen Method for Calculating Potential

Input: A mesh of elements e_j and a group of point charges Q_i
Return: Potential at locations of charges
for each charge Q_i {charge assignment, Section 3.1}
  place Q_i in element
  solve for screen {(3.10) or (3.19)}
for each element e_j
  if element $e_j \in$ surface
    adjust boundary conditions as necessary {see (3.26)}
  apply charge assignment operator to form ρ_{m} {see (3.9)}
perform multigrid solve of $-\nabla^2\Phi^{\mathrm{sm}} = \rho_{\mathrm{m}}$ {see (3.2)}
for each charge Q_i {evaluations, Sections 3.1 and 3.1}
  $\Phi_i \leftarrow \Phi_i^{\mathrm{sr}} + \Phi_i^{\mathrm{sm}}$ {mesh-to-charge assignment}

We assume a collection of N charges $\mathcal{Q} \equiv \{Q_i\}_{i=1}^N$ located at $\mathbf{x}_i^{\mathrm{c}}$ in a cube $\Omega = [0, L]^3$. A mesh with $N_{\mathrm{el}} = n_{\mathrm{el}}^x \times n_{\mathrm{el}}^y \times n_{\mathrm{el}}^z$ elements is constructed to conform to the domain, and a uniform mesh is assumed in each direction for simplicity of presentation — i.e., $n_{\mathrm{el}} = n_{\mathrm{el}}^x = n_{\mathrm{el}}^y = n_{\mathrm{el}}^z$. Finally, the collocation points for q-order basis functions on the mesh are denoted $\mathbf{x}_j^{\mathrm{m}}$, with $j = 1, \ldots, M \equiv (qn_{\mathrm{el}} + 1)^3$.

Charge assignment

A central component of particle-mesh methods is the assignment of singular charges to the mesh, yielding a mesh-based charge density function, $\rho_{\mathrm{m}}(\mathbf{x}^{\mathrm{m}})$. In particular, we seek an assignment function $\mathcal{W}(\mathbf{x})$ that reflects our specially-selected screen functions and provides a weighting that distributes a charge Q_i at $\mathbf{x}_i^{\mathrm{c}}$ to each collocation point $\mathbf{x}_j^{\mathrm{m}}$ of the basis functions:

$$\rho_{\mathrm{m}}(\mathbf{x}_j^{\mathrm{m}}) = \sum_{i=1}^{N} Q_i \mathcal{W}(\mathbf{x}_j^{\mathrm{m}}; \mathbf{x}_i^{\mathrm{c}}). \tag{3.3}$$

Existing methods use Lagrange polynomials (PME [6]) or B-splines (P^{3}M [5] and SPME [7]) for this weighting, where B-splines work particularly well with FFTs. The charge assignment function impacts both accuracy and efficiency of the method. In our approach we design an assignment operator based directly on polynomial basis functions for compatibility with a finite-element-based Poisson solver.

Defining the polynomial screens

We define our screen density function for a single charge $Q_i \in \mathcal{Q}$ as

$$\rho_i(\mathbf{x}) = \sum_j c_j \psi_j(\mathbf{x}), \tag{3.4}$$

with linear superposition providing the extension to multiple charges. As suggested by the form of (3.3), we will consider unit charges in our rules for screen construction (defining $\mathcal{W}(\mathbf{x})$), with the result then scaled by any charge strength Q. Here $\psi_j(\mathbf{x})$ are a collection of q-order Lagrange basis functions over an index set determined as follows. If charge Q_i is located within element τ_j of the mesh, we choose $V_\rho^i = \cup_{\tau \cap \tau_j \neq 0} \tau$ to be the interpolation support of the charge assignment operator. That is, the support is the union of the element of the mesh that includes the charge along with all neighboring elements, leading to a support of 27 elements in three dimensions. Generalization to other choices for this support is briefly discussed in Section 3.4. To construct the polynomial screen, we consider the degrees of freedom which are interior to or on the faces of the element containing the charge. For q-order interpolating polynomials, this leads to $\dim(V_\rho^i) = (q+1)^3$ degrees of freedom. These degrees of freedom are determined so that the charge-screen combination has a potential that decays rapidly in space by considering the multipole expansion of the screen for a point well outside the screen, given by

$$\Phi_i^{\text{sc}}(\mathbf{x}) = \frac{1}{\hat{R}} \int_{V_\rho^i} \rho_i(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi} - \frac{1}{\hat{R}^2} \int_{V_\rho^i} \rho_i(\boldsymbol{\xi})\, (\boldsymbol{\xi} \cdot \hat{\mathbf{r}})\, \mathrm{d}\boldsymbol{\xi} + \frac{1}{2\hat{R}^3} \int_{V_\rho^i} \rho_i(\boldsymbol{\xi}) [3(\boldsymbol{\xi} \cdot \hat{\mathbf{r}})^2 - |\boldsymbol{\xi}|^2]\, \mathrm{d}\boldsymbol{\xi} + \cdots \tag{3.5}$$

where $\hat{\mathbf{R}} = (\mathbf{x} - \mathbf{x}_i)/h$, where $\mathbf{x}$ is the observation point, $\mathbf{x}_i$ is the charge location, h is the mesh size, and $\hat{R} = |\hat{\mathbf{R}}|$. The quantity $\hat{\mathbf{r}}$ is the unit direction vector $\hat{\mathbf{R}}/\hat{R}$.

For a charge Q_i located at $\mathbf{x}_i = (x_i, y_i, z_i)$ in element τ_j (in the interest of readability, we drop superscript "c" when it is clear that the reference is to a charge location), we denote the offset $\boldsymbol{\delta}_i = (\delta_i^x, \delta_i^y, \delta_i^z) = \mathbf{x}_i - \bar{\mathbf{x}}_j$ with respect to the center of the element $\bar{\mathbf{x}}_j$, and define the (l,m,n)-moment and *centered* (l,m,n)-moment of the screen function as

$$\rho_i^{(l,m,n)} = \int_{V_\rho^i} (x - \delta_i^x)^l (y - \delta_i^y)^m (z - \delta_i^z)^n \rho_i(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{3.6}$$

$$\bar{\rho}_i^{(l,m,n)} = \int_{V_\rho^i} x^l y^m z^n \rho_i(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{3.7}$$

If we take the origin as the element center $\bar{\mathbf{x}}_j$, dividing $\rho_i^{(0,0,0)} = \int_{V_\rho^i} \rho_i(\mathbf{x})\, \mathrm{d}\mathbf{x}$ by $\hat{R} = |\mathbf{x} - \mathbf{x}_i|/h$ gives the first term of the screen's multipole expansion from (3.5). Thus, requiring $\rho_i^{(0,0,0)} = 1$ (for a unit charge)

guarantees that the combined point-charge and screen have a potential that decays at least as fast as $1/R^2$ with distance from the point charge. Likewise, zeroing higher moments of the screen enforces the cancellation of dipole and higher-order terms, making the far-field behavior of the screen follow the behavior of a point charge by increasing its symmetry. This accelerates the long-range decay rate of the point-charge-and-screen combination and reduces the number of interactions that must be explicitly represented by point-to-point computations. With the available degrees of freedom, a screen of order q cancels all terms up to $R^{-(q+1)}$, leaving $\Phi^{\text{sr}} = 1/R - \Phi^{\text{sc}} \sim R^{-(q+2)}$ for a unit charge. This is summarized in Table 3.1, which shows the moments that result from performing the vector operations in the integrands of (3.5).

Power of R	Single terms	Mixed terms
R^{-1}	1	—
R^{-2}	x, y, z	—
R^{-3}	x^2, y^2, z^2	xy, xz, yz
R^{-4}	x^3, y^3, z^3	$x^2y, xy^2, x^2z, xz^2, y^2z, yz^2, xyz$
$\vdots$	$\vdots$	$\vdots$
R^{-N}	$x^{N-1}, y^{N-1}, z^{N-1}$	$x^l y^m z^n$, with $1 \leq l,\, m,\, n \leq N-2$, and $l+m+n = N-1$

Table 3.1: Polynomial terms in multipole expansion.

Constructing the screen

The goal is to perform multipole cancellations with screens that are also compatible with the basis functions of our finite element discretization. With this description, each screen is composed of $N_{\text{sc}} = n_{\text{sc}}^3 = (q+1)^3$ nodal screen basis functions, ψ:

$$\rho_i(\mathbf{x}) = \sum_{j=0}^{N_{\text{sc}}-1} c_j \psi_j(\mathbf{x}). \tag{3.8}$$

Thus, revisiting (3.3), the assignment operator $\mathcal{W}$ is

$$\mathcal{W}(\mathbf{x}_j^{\text{m}}; \mathbf{x}_i^{\text{c}}) = \begin{cases} \sum_{k=0}^{N_{\text{sc}}-1} c_k(\boldsymbol{\delta}_i)\psi_k(\mathbf{x}_j^{\text{m}}) & \mathbf{x}_j^{\text{m}} \in V_\rho^i \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

Restricting the first N_{sc} moments leads to a $N_{\text{sc}} \times N_{\text{sc}}$ linear system for the coefficients $\mathbf{c}$ in (3.8):

$$\begin{bmatrix} \psi_0^{(0,0,0)} & \psi_1^{(0,0,0)} & \cdots & \psi_{N_{\text{sc}}-1}^{(0,0,0)} \\ \psi_0^{(1,0,0)} & \psi_1^{(1,0,0)} & \cdots & \psi_{N_{\text{sc}}-1}^{(1,0,0)} \\ \vdots & & \ddots & \vdots \\ \psi_0^{(q,q,q)} & \psi_1^{(q,q,q)} & \cdots & \psi_{N_{\text{sc}}-1}^{(q,q,q)} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N_{\text{sc}}-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{3.10}$$

where ${\psi_j}^{(l,m,n)}$ is the (l,m,n)-moment, as defined in (3.6) for ρ, of the j-th screen basis function $\psi_j(\mathbf{x})$.

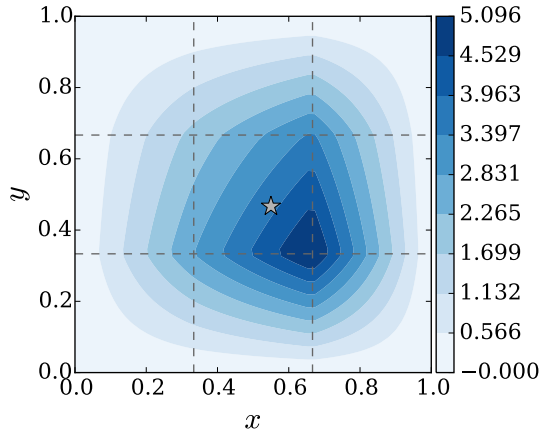
Figure 3.1 shows cross-sections of example screens constructed using $q = 1, \ldots, 4$. Each screen's peak is attained near the marked charge, and the screens are constructed to decay to zero at the edge of V_ρ. The screens have support in the active screen region V_ρ^i and for $q > 1$, the screens are in general non-monotone.

We confirm in Figure 3.2 that screens constructed in this fashion yield potentials with the expected behavior: in all cases, the far-field behavior approximates $1/R$. In Figure 3.3, we estimate mean and peak errors incurred for point-to-point interaction truncation at a distance $\hat{R}_c = R_c/h$. To do this, the short-range potential is constructed for $N = 84$ charge locations within an element and sampled in 42 directions; the behavior of $|\Phi^{\text{sr}}|$ is shown in the figure as the weighted average $|\Phi^{\text{sr}}|_{\text{avg}}$ and the maximum from all samples $|\Phi^{\text{sr}}|_{\max}$. In addition, the radius from a charge location is also normalized as $\hat{R} = R/h$, and is denoted $\hat{R}$. We see that twice the screen-size scale $\hat{R} \approx 3$ corresponds to the expected start of the asymptotic decay behavior. This is the distance at which a multipole expansion is generally considered "well-separated" and expected to show convergence with R. Both the mean and peak errors show the expected behavior for increasing q beyond this distance.

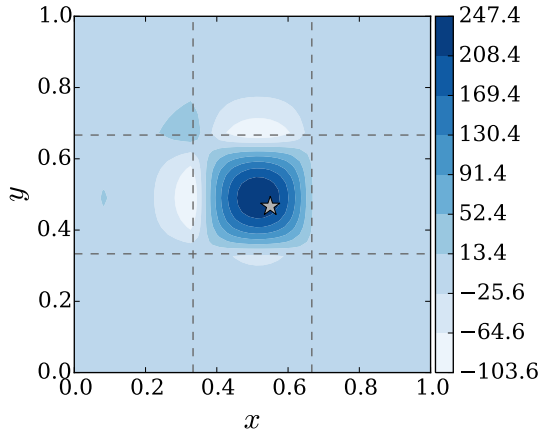
A fast algorithm for screen construction

Solving (3.10) directly requires $\mathcal{O}(N_{\text{sc}}^3)$ operations for each screen, which is feasible, but is not necessary in general. In the following, we design a fast algorithm for computing screens, which follows from a generalization of the Parallel Axis Theorem applied to moments used in the system. First, we note that the moments are additive. For example, the first moment in variable x of basis function ψ_j satisfies

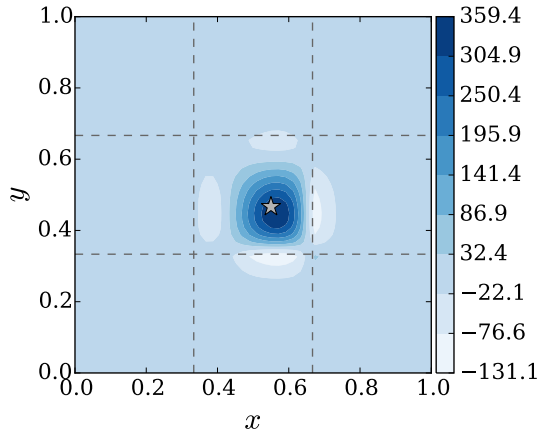
$$\begin{aligned} \psi_j^{(1,0,0)} &= \int_{V_\rho} x\psi(\mathbf{x}) \, \mathrm{d}\mathbf{x} - \delta_i^x \int_{V_\rho} \psi(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &= \bar{\psi}_j^{(1,0,0)} - \delta_i^x \psi_j^{(0,0,0)}, \end{aligned} \tag{3.11}$$



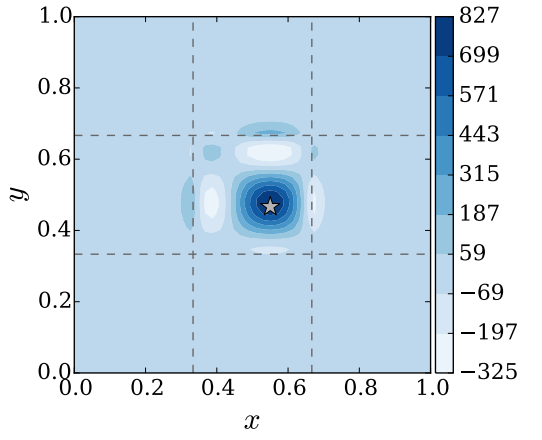
(a) $q = 1$



(b) $q = 2$

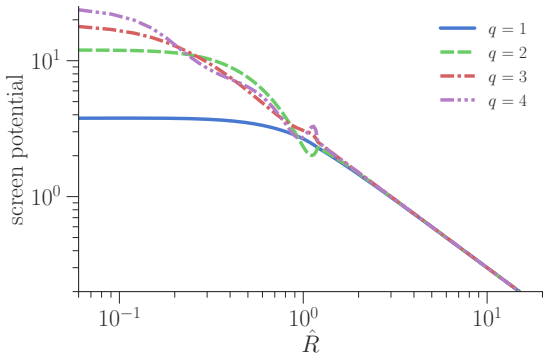


(c) $q = 3$

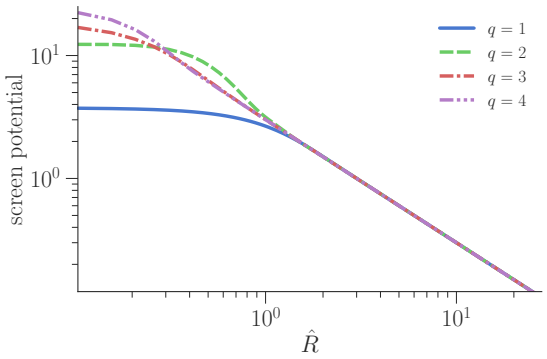


(d) $q = 4$

Figure 3.1: Example of linear through quartic screens for a single charge (marked by a star).



(a)



(b)

Figure 3.2: Screen potential vs $\hat{R} = |\mathbf{x}-\mathbf{x}_i|/h$ in two directions: (a) $\mathbf{x}-\mathbf{x}_i \propto (1,0,0)$ and (b) $\mathbf{x}-\mathbf{x}_i \propto (1,1,1)$.

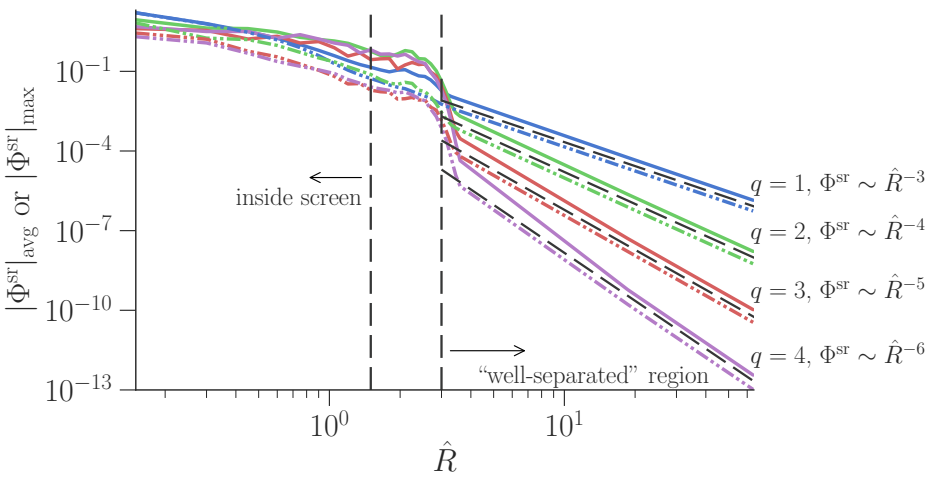


Figure 3.3: Maximum (solid) and average (dash-dot-dot) short-range potentials for screens with $q = 1$ to 4. The straight dash-dot lines denote the expected slopes.

where $\bar{\psi}_j{}^{(1,0,0)}$ is the *centered* moment of the j-th basis function as in (3.7). Therefore, second row of (3.10) is equivalent to

$$\sum_{j=0}^{N_{\mathrm{sc}}-1} c_j \bar{\psi}_j^{(1,0,0)} = \delta_i^x \sum_{j=0}^{N_{\mathrm{sc}}-1} c_j \psi_j^{(0,0,0)} = \delta_i^x. \tag{3.12}$$

Similarly, the second moment of ψ_j satisfies

$$\begin{aligned} \psi_j^{(2,0,0)} &= \int_{V_\rho} x^2 \psi(\mathbf{x})\, \mathrm{d}\mathbf{x} - 2\delta_i^x \int_{V_\rho} x\psi(\mathbf{x})\, \mathrm{d}\mathbf{x} + (\delta_i^x)^2 \int_{V_\rho} \psi(\mathbf{x})\, \mathrm{d}\mathbf{x} \\ &= \bar{\psi}_j^{(2,0,0)} - 2\delta_i^x \bar{\psi}_j^{(1,0,0)} + (\delta_i^x)^2 \psi_j^{(0,0,0)}, \end{aligned} \tag{3.13}$$

so the third row of (3.10) becomes

$$\begin{aligned} \sum_{j=0}^{N_{\mathrm{sc}}-1} c_j \bar{\psi}_j^{(2,0,0)} &= 2\delta_i^x \sum_{j=0}^{N_{\mathrm{sc}}-1} c_j \bar{\psi}_j^{(1,0,0)} - (\delta_i^x)^2 \sum_{j=0}^{N_{\mathrm{sc}}-1} c_j \psi_j^{(0,0,0)} \\ &= 2(\delta_i^x)^2 - (\delta_i^x)^2 = (\delta_i^x)^2. \end{aligned} \tag{3.14}$$

Continuing this procedure for other moments in (3.10) yields:

$$\begin{bmatrix} \psi_0^{(0,0,0)} & \psi_1^{(0,0,0)} & \dots & \psi_{N_{\rm sc}-1}^{(0,0,0)} \\ \bar{\psi}_0^{(1,0,0)} & \bar{\psi}_1^{(1,0,0)} & & \\ \vdots & & & \vdots \\ \bar{\psi}_0^{(q,0,0)} & & \ddots & \bar{\psi}_{N_{\rm sc}-1}^{(q,0,0)} \\ \bar{\psi}_0^{(0,1,0)} & & & \\ \vdots & & & \vdots \\ \bar{\psi}_0^{(q,q,q)} & & \dots & \bar{\psi}_{N_{\rm sc}-1}^{(q,q,q)} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \\ \vdots \\ \\ \\ c_{N_{\rm sc}-1} \end{bmatrix} = \begin{bmatrix} 1 \\ \delta_i^x \\ \vdots \\ (\delta_i^x)^q \\ \delta_i^y \\ \vdots \\ (\delta_i^x)^q(\delta_i^y)^q(\delta_i^z)^q \end{bmatrix}, \tag{3.15}$$

which we write compactly as $\mathcal{C}\mathbf{c} = \mathbf{f}$. An advantage of this form is that for a uniform mesh, the matrix $\mathcal{C}$ is the same for each screen, since the moments reference the center of the element. As a result, the matrix can be pre-factorized, leading to a complexity of only $\mathcal{O}(N_{\rm sc}^2)$ to solve for each screen. Furthermore, this reduces set-up cost by eliminating the need to calculate the moment integral entries of the matrix in (3.10) for every screen.

For small q this yields a small operation count, yet the computation is further reduced if $\psi(\mathbf{x})$ is separable, as is the case for a regular cubic mesh shown. In this case,

$$\psi_\kappa(\mathbf{x}) = \omega_i(x)\omega_j(y)\omega_k(z), \tag{3.16}$$

where ω_i are the one-dimensional nodal basis functions for a mesh size h and $\kappa = i + (q+1)j + (q+1)^2k$ with $i, j, k \in [0, q]$. With ψ separable, the moment integrals are also separable:

$$\begin{aligned} \bar{\psi}_\kappa^{(l,m,n)} &= \int_{V_\rho^i} x^l y^m z^n \psi_\kappa(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \left(\int_{-3h/2}^{3h/2} x^l \omega_i(x)\,\mathrm{d}x\right)\left(\int_{-3h/2}^{3h/2} y^m \omega_j(y)\,\mathrm{d}y\right)\left(\int_{-3h/2}^{3h/2} z^n \omega_k(z)\,\mathrm{d}z\right). \end{aligned} \tag{3.17}$$

Following the notation of (3.7), we define

$$\bar{\omega}_i^{(l)} = \int_{-3h/2}^{3h/2} x^l \omega_i(x)\,\mathrm{d}x \tag{3.18}$$

and likewise for the one-dimensional y and z centered moments. We take $c_\kappa = w_i^x w_j^y w_k^z$ and recognize that the right-hand side of (3.15) is also separable as $f_\mu = (\delta^x)^l (\delta^y)^m (\delta^z)^n$ with $\mu = l + (q+1)m + (q+1)^2 n$. This yields three equivalent $(q+1)^2$ systems of the form

$$\sum_{i=0}^{q} \bar{\omega}_i^{(l)}(x) w_i^x = (\delta^x)^l, \quad l = 0, \ldots, q \tag{3.19}$$

which are solved independently. The inverse of the matrix is computed once and applied for all right hand sides, resulting in only $\mathcal{O}(n_{\mathrm{sc}}^2)$ operations per screen. This method also extends to regular rectangular meshes, where h is not necessarily equal in each direction.

Solution of the mesh potential

Given our construction of ρ_m through the screens ρ_i built from the finite element basis functions via (3.9), the solution of Φ^{sm} is straightforward. Since the screens are polynomial of degree q, we use a finite element solver with basis functions u^{m} of order $p = q+2$, which results in a symmetric, positive definite sparse linear system (under realistic assumptions regarding the boundary conditions).

Multigrid preconditioners are effective for this problem, even for high-order bases, and allow the sparse matrix problem to be solved to any level of accuracy. As an example, consider the case of a high-order finite element discretization of the Poisson problem with Dirichlet boundary conditions. Figure 3.4 shows the convergence history of a multigrid preconditioned conjugate gradient method for basis functions of order $p = 1$ through 6. An algebraic multigrid preconditioner, based on smoothed aggregation using a more general strength measure [46] and optimal interpolation operator [47], is used. We observe only a weak dependence on p. Moreover, more advanced multigrid techniques have shown still better scalings for both Poisson and other elliptic problems such as for Stokes flow [48, 49]. Importantly, multigrid preconditioners are also well-known to exhibit high parallel efficiency [41, 50]. In the following tests, we use AMG through the BoomerAMG package [51].

Evaluation of the smooth potential at charge locations

The next step is to evaluate Φ^{sm} at the charge locations. For standard $\mathrm{P^3M}$ and PME implementations, this involves interpolation with the Lagrangian or B-spline basis functions from the charge assignment. In

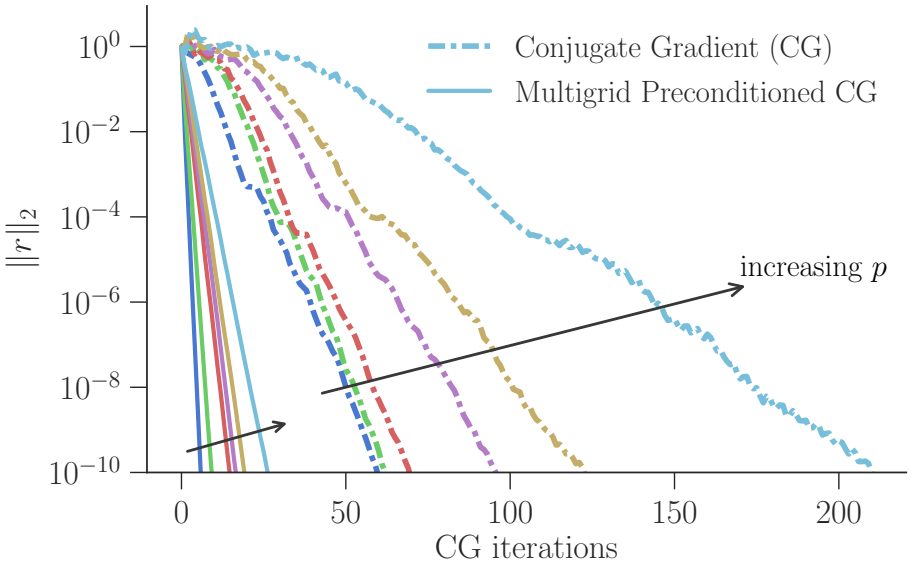


Figure 3.4: Convergence history for the Poisson problem using basis elements of order $p = 1, \ldots, 6$ for both the conjugate gradient method (dot-dashed) and multigrid preconditioning (solid).

contrast, our method requires no interpolation, though interpolation can be used to speed up calculations, if desired. Since the smooth potential exists in each element as a linear combination of coefficients — i.e., the values of Φ^{sm} at $\mathbf{x}^{\mathrm{m}}$ for all N_{m} points in an element — and the basis functions u^{m}, the smooth potential is expressed at any point as

$$\Phi^{\mathrm{sm}}(\mathbf{x}) = \sum_{j=1}^{N_{\mathrm{m}}} \Phi^{\mathrm{sm}}(\mathbf{x}_j^{\mathrm{m}}) u_j^{\mathrm{m}}(\mathbf{x}), \tag{3.20}$$

where N_{m} is the number of collocation points in an element. Direct evaluation at the charge locations $\mathbf{x} = \mathbf{x}_i^{\mathrm{c}}$ is straightforward.

Short-range potential

Our formulation for the mesh-based potential yields a more complex short-range interaction than PME. In addition to R, the short-range interaction now also depends on the position of the charge relative to the underlying mesh. Consequently, additional effort is required to evaluate the short-range interaction. However, the calculation is *local*, which is an added benefit in a manycore setting.

The short-range potential at point $\mathbf{x}$ due to a charge Q_i located at $\mathbf{x}_i$ is

$$\Phi_i^{\mathrm{sr}}(\mathbf{x}) = \frac{Q_i}{|\mathbf{x} - \mathbf{x}_i|} - \Phi_i^{\mathrm{sc}}(\mathbf{x}), \tag{3.21}$$

where

$$\Phi_i^{\mathrm{sc}}(\mathbf{x}) = Q_i \int_{V_\rho^i} \frac{\rho_i(\boldsymbol{\xi})}{|\mathbf{x}-\boldsymbol{\xi}|} \mathrm{d}\boldsymbol{\xi}. \tag{3.22}$$

Recall that Φ^{sm}, the smooth potential, includes the sum of all individual screen potentials Φ_i^{sc} and the boundary effects (here, the periodic images of the charges). Though feasible, performing accurate quadrature for each screen individually is computationally expensive. We therefore shift a significant portion of this computational effort to a pre-processing step, for which there are multiple options.

One approach is to consider a look-up table of pre-computed values for the screen potential evaluated at $\mathbf{x}_j$ due to a charge at $\mathbf{x}_i^{\mathrm{c}}$. These values comprise a six-dimensional look-up table, $\Phi^{\mathrm{sc}}(\mathbf{x}_j - \mathbf{x}_i^{\mathrm{c}}; \boldsymbol{\delta}_i)$, since they are a function of the difference between the evaluation point and the charge location, and also the offset of the charge within its element (which determines the screen).

With some additional computation, but still without resorting to direct evaluation of (3.22), it is possible to remove the charge offset interpolation to reduce errors. We accomplish this by recognizing the screen's formulation as a linear combination of basis functions,

$$\Phi_i^{\mathrm{sc}}(\mathbf{x}) = Q_i \sum_{j=0}^{N_{\mathrm{sc}}-1} c_j(\boldsymbol{\delta}_i) \int_{V_\rho^i} \frac{\psi_j(\boldsymbol{\xi})}{|\mathbf{x}-\boldsymbol{\xi}|} \mathrm{d}\boldsymbol{\xi} \tag{3.23}$$

$$= Q_i \sum_{j=0}^{N_{\mathrm{sc}}-1} c_j(\boldsymbol{\delta}_i) \Phi_j^{\mathrm{b,sc}}(\mathbf{x} - \mathbf{x}_i^{\mathrm{c}}). \tag{3.24}$$

This approach yields N_{sc} look-up tables for basis-function potential values $\Phi^{\mathrm{b,sc}}(\mathbf{x}_j - \mathbf{x}_i^{\mathrm{c}})$. However, for $q \geq 2$ the polynomial nature of the screen leads to non-monotonic decay for some directions within the region where the screen is active, as shown in Figure 3.2. Consequently, a direct implementation of a look-up table for such functions requires sufficient resolution, which is harder to achieve for larger q. In contrast to Ewald-based methods, where short-range computations are efficient by design, the cost of these look-up tables may be non-trivial. Therefore, for good performance, knowledge of the underlying structure of the screen potentials should be used to inform both the storage locations for the look-up table values and the interpolation method.

A note about the self term

If the point $\mathbf{x}$ is the location of a charge, we do not wish to include the potential due to this charge in our calculation. However, we do still need to subtract the screen potential from the charge's own screen, which is included in Φ^{sm}; this is sometimes called the "self" term. We can allow for this by amending our short-range

potential expression to include both cases:

$$\Phi_i^{\mathrm{sr}}(\mathbf{x}) = \begin{cases} \frac{Q_i}{|\mathbf{x}-\mathbf{x}_i|} - \Phi_i^{\mathrm{sc}}(\mathbf{x}) & |\mathbf{x}-\mathbf{x}_i| > 0 \\ -\Phi_i^{\mathrm{sc}}(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{3.25}$$

Alternative boundary conditions

The formulation above is presented under the assumption of periodic boundary conditions, which is the simplest case and important for a range of applications. It is straightforward to generalize boundary conditions via the mesh potential Φ^{sm}. This is accomplished by adjusting for short-range effects present at the boundary and then proceeding in the usual manner for a finite element problem with the given type of boundary conditions. For example, for a Dirichlet boundary condition of $\Phi = g$ on ∂V, the condition for our mesh problem becomes

$$\Phi^{\mathrm{sm}}|_{\partial V} = g - \Phi^{\mathrm{sr}}|_{\partial V}, \tag{3.26}$$

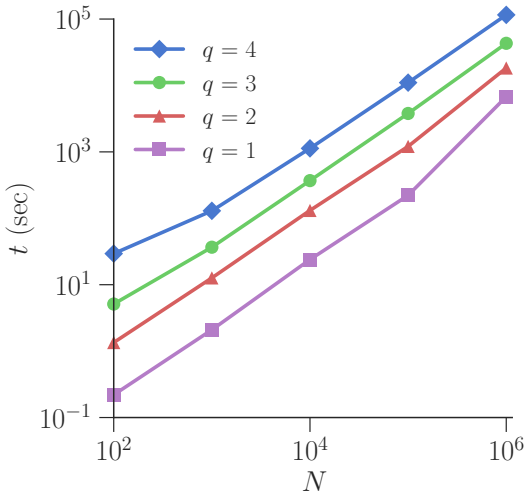
which leads to

$$\Phi|_{\partial V} = \Phi^{\mathrm{sm}}|_{\partial V} + \Phi^{\mathrm{sr}}|_{\partial V} = g. \tag{3.27}$$

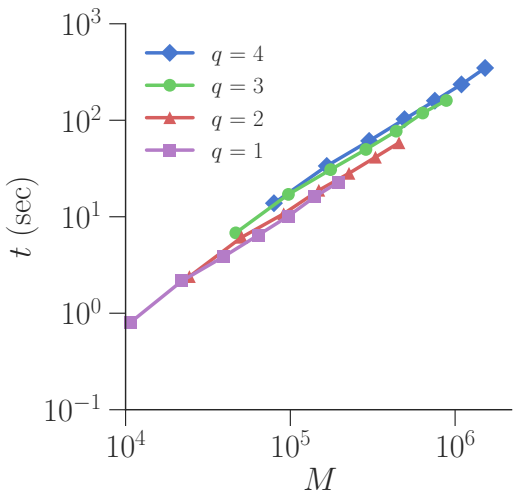
A similar approach is used in [32]. This also extends to the case of Neumann or mixed-type boundary conditions, with the usual constraint to address the non-uniqueness of the fully Neumann problem. However, free-space conditions require careful treatment, as in any mesh-based discretization scheme.

3.2 Performance model

The computational cost of the method for N charges is formulated as $\mathcal{O}(N)+\mathcal{O}(M)$, where $M = (pn_{\mathrm{el}}+1)^3$ is the total number of degrees of freedom in the mesh. Example CPU time scalings for the major N-related components is illustrated in Figure 3.5a, with the M-dependent mesh solve times shown in Figure 3.5b. Given N and M and assuming on average $N_{\mathrm{el}}^{\mathrm{sr}}$ neighboring elements in the short-range interaction list for each charge, then it is possible to express the coefficients in the linear $\mathcal{O}(N) + \mathcal{O}(M)$ operation count in terms of p. Such a formulation provides a more detailed description of the actual costs of each component of the method and their relationships to the order of the screens.



(a) Total CPU time vs. N for the main N-dependent components of the algorithm, including screen creation, short-range calculation, basis function evaluation, and combination of short-range and mesh potentials at charge locations.



(b) CPU time to solve for mesh potential versus number of mesh points M. The number of elements per coordinate direction is varied from 7 to 19.

Figure 3.5: Total CPU time for (a) N-related components, and (b) mesh solve (M-related).

Breakdown of costs

Screens $\rho_i(\mathbf{x})$ of order q are built out of $(q+1)^3 = N_{\text{sc}}$ basis functions — recall that $p = q + 2$. The corresponding finite element solve associated with these screens involves $(p+1)^3$ degrees of freedom per element and a total of M degrees of freedom. We also define the average number of charges per element as $\tilde{N} = N/N_{\text{el}}$.

Screen construction

For each evaluation, the element containing each charge is identified, and the offsets from the center of these elements determined. This incurs a small $\mathcal{O}(N)$ cost, which we designate $C_1 N$. The screen coefficients are then calculated. As shown in (3.19), assuming pre-computed inverses, this amounts to three matrix-vector multiplications of size $q+1 = p-1$, for a cost of $6(p-1)^2 - 3(p-1)$. We then multiply the one-dimensional weights, resulting in two additional floating point multiplications. The total cost for determining the screen coefficients is thus

$$\sim [6(p-1)^2 - 3(p-1) + 2(p-1)^3]N. \qquad \text{(screen construction)}$$

Short-range potential

The cost of evaluating the short-range potential depends on the method chosen for calculating Φ^{sc}, as discussed in Section 3.1. In addition, there is a cost of $\mathcal{O}(N)$ due to the singular part of the short-range calculations, which we denote SN. For a general six-dimensional look-up table, the cost of calculating Φ^{sc} at a point due to all charges in the short-range interaction volume is $C_2\tilde{N}N_{\text{el}}^{\text{sr}}$, where C_2 depends on the order of interpolation used. If N_{sc} three-dimensional look-up tables are used, as we have done in the example calculations of Section 3.3, then the interpolation is repeated for $(q+1)^3$ tables and combined by an inner product with the screen coefficients and a multiplication by Q_i for a total of $[C_2(p-1)^3 + 2(p-1)^3]\tilde{N}N_{\text{el}}^{\text{sr}}$. The cost for the short-range calculation is then

$$\sim SN + [C_2(p-1)^3 + 2(p-1)^3]\tilde{N}N_{\text{el}}^{\text{sr}}N. \qquad \text{(point-to-point evaluation)}$$

Since $\tilde{N} = N/N_{\text{el}}$, this expression is also written in terms of N^2. However, we assume that in practice, $\tilde{N}N_{\text{el}}^{\text{sr}}$ is chosen to be small enough to render this effectively as $\mathcal{O}(N)$. Furthermore, if $\tilde{N}$ is > 1, this cost is reduced further by calculating the effects of all charges in an element at once in an "element-to-point" operation. To do this, we compile a combined list of $Q_i\mathbf{c}_i$ for all the charges in any given element, so that the screen potential for this sum at a point as calculated by (3.24) is the same as if the charges were handled individually. The cost then is then reduced by a factor of $\tilde{N}$ yielding

$$\sim SN + [C_2(p-1)^3 + 2(p-1)^3 - 1]N_{\text{el}}^{\text{sr}}N. \qquad \text{(element-to-point evaluation)}$$

Mesh solve

The "transfer" of the order-q screens to a representation in order $p = q+2$ basis functions by (3.9) to construct the source ρ_{m} in the right-hand side of the finite element solve (3.2) requires an inner product between a vector containing the screen coefficients $\mathbf{c}(\boldsymbol{\delta})$ with the evaluation of the order-q basis functions at the collocation points, followed by a multiplication by Q_i. This is done at each degree of freedom within an active screen area, for a total of $(3p+1)^3 \times [2(p-1)]N$ operations. The multigrid solve for the finite element problem is $\mathcal{O}(M)$, with a coefficient C that depends on the convergence of the iterations, but is considered low in practice. Overall the mesh solve thus has complexity

$$\sim \{(3p+1)^3 \times [2(p-1)^3]\}N + CM. \qquad \text{(mesh solve)}$$

Evaluation

The smooth potential is written as a combination of basis functions at the location of each charge, as in (3.20). Thus evaluation involves $(p+1)^3$ basis functions at a cost of $2p$ operations for each function. However, empirically we find that this cost is minimal in terms of CPU time.

Summary

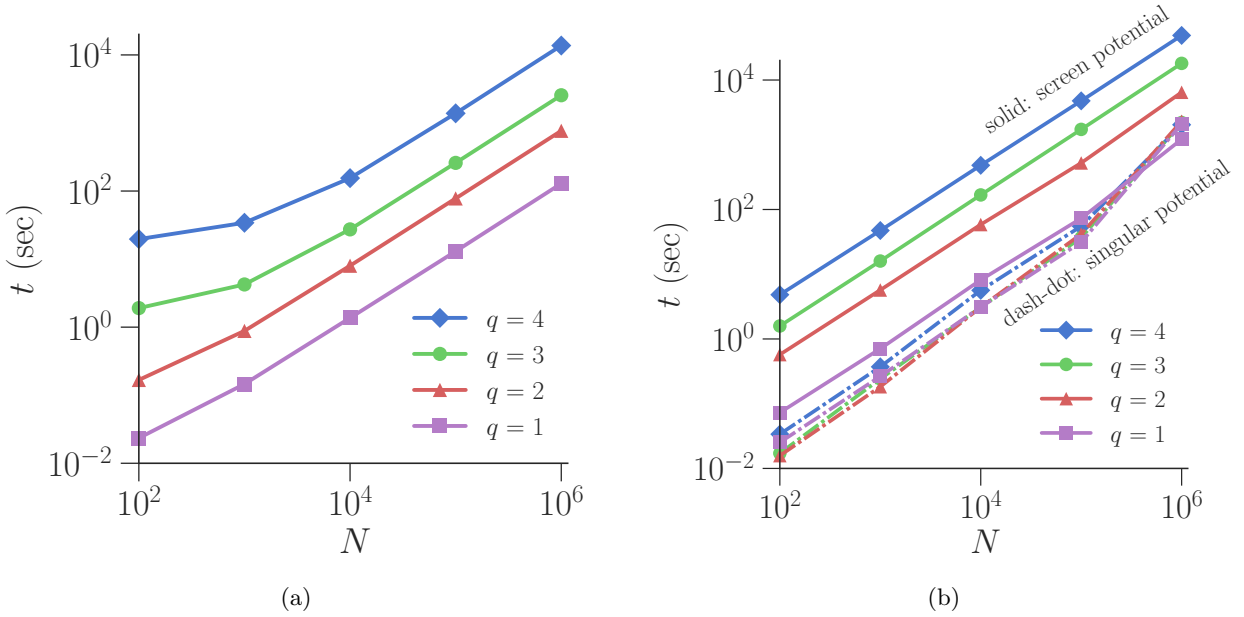
The screen creation (mostly due to the "transfer" portion) and short-range interaction calculations are the most costly even for modest values of p given the scaling shown above. The relative costs of these two portions of the algorithm depend on choices in short-range calculation method, mesh size, and q. For any given cutoff error, decreasing mesh spacing decreases the number of short-range interactions, but results in an increased number of collocation points M in the mesh solve. Likewise, increasing q also decreases the number of short-range interactions, but at the price of the increased cost of constructing and manipulating screens for larger q. Calculating the short-range effects of each individual charge becomes more costly with increased q, though at a slower rate than the transfer. The scaling of these components is shown in Figure 3.6 for cases of $N = 10^2$ to 10^6 randomly distributed particles in a triply-periodic box with 6 859 elements. The tests were done using a dual, quad-core Intel Xeon E5506 CPU with 48 GB of main memory. It is noted that once $\tilde{N} \gg 1$, the singular short-range calculation loses its linearity in N. However, the screen potential portion of the short-range calculation retains its linearity due utilization of the "element-to-point" evaluation method.

3.3 Example calculations

Periodic test case

We first consider cases with N ranging from 10^2 to 10^5 unit charges placed in a triply-periodic unit cube of elements with $h = 0.06\bar{6}$. The exact positions are selected randomly, but distributed so that any given charge experiences both long-range interactions, on the scale of the overall periodic domain size, and short-range interactions of comparable magnitude. This is done to provide a balanced test of both the short-range and smooth portions of our decomposition. To achieve this, the charges are randomly distributed within two smaller cubes: $[0, 1/2]^3$ is biased toward positive charges, 55% to 45%, and $[1/2, 1]^3$ is biased equally strongly toward negative charges. This set-up is visualized in Figure 3.7a for $N = 100$.

The potential is then calculated using a short-range interaction of $7 \times 7 \times 7 = 343$ elements (corresponding



(a) (b)

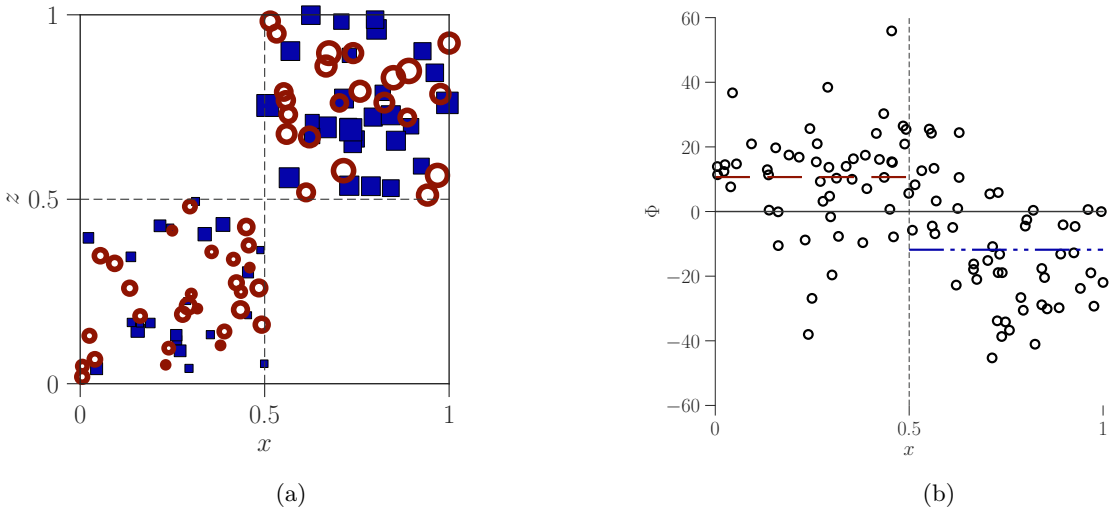
Figure 3.6: Example CPU time vs. N for the two most costly local portions of the algorithm: (a) creation of the screens, (b) calculation of short-range interactions for $N_{\text{el}}^{\text{sr}} = 7 \times 7 \times 7$. The dot-dash lines show the time associated with calculating singular potentials, while solid lines show times calculating element-to-point screen potentials. At large N, there is an expected breakdown in linearity for the singular potential calculations.

to a minimum possible value of 3 for the cutoff distance $\hat{R}_c$) for linear through quartic screens. This short-range cutoff is chosen to ensure that the short-range potential of every charge near the cutoff exhibits asymptotic behavior. The short-range calculation uses the approach of (3.24), with N_{sc} look-up tables.

Remark 3.1. *In our current implementation, we use a variation (but equivalent form) to this construction, in which the values stored in the tables are for "basis screens" instead of screen basis functions. These basis screens, ρ^{basis}, are the polynomial screens associated with each node in an order-q finite element. The values of each table are computed as a Dirichlet finite element solution for Poisson's equation, with $-\nabla^2\Phi_i^{basis} = \rho_i^{basis}$. The computation is completed in a domain larger than the size that will be kept in the look-up table to minimize boundary effects. Because these basis screens follow our moment-canceling rules, they have long-range decay $\sim \hat{R}^{-(q+2)}$, and the boundary conditions are accurately set by the first terms of the multipole expansion (3.5). The finite element solver uses basis functions of order p, and the look-up tables are stored in terms of their order-p basis functions, allowing them to be evaluated and combined in the same way as Φ^{sm} for all charge locations. We note that because the number of tables and coefficients is unchanged, the computational complexity for the short-range calculation is not altered by this variation.*

Upon calculation the potential is compared, allowing for a constant which is included in a potential and in this case is equal to the average value of Φ^{sm} throughout the computational domain, with that of an

Ewald summation Φ^{E} with large enough resolution that we consider it the "exact" solution. This uses two periodic images in physical space with $a^2 = 6.25$ and four modes for each direction in the Fourier sum. As we see for a representative calculation in Figure 3.8, the method has super-algebraic convergence with q. It is important to note that convergence in this case with respect to mesh spacing, h, is not easily predicted. Indeed, since decreasing h with a fixed number of charges N results in few charges per element (on average), the short-range error has fewer charges contributing to its total error in the case of a mesh based screen. Similarly, extending the short-range interaction more broadly would result in little impact from changing h.



(a) (b)

Figure 3.7: Configuration of demonstration calculation: (a) Location of ∘ positive and □ negative charges for $N = 100$, with a marker's size indicating the y-location of the charge, and (b) Φ at each charge location for the 100 charge case, arranged by the charge's location in x; the red dashed and blue dash-dot lines mark the average potential value at locations in the positively-biased group and negatively-biased group, respectively.

In each of these tests, we use an algebraic multigrid preconditioned GMRES solver with single precision residual tolerance — i.e., $1e-7$. Boomeramg [51] is used and the resulting method yields 8 or fewer iterations in each of the tests reported above. The timing dependence on mesh size is reported in Figure 3.5b where we see that the solver exhibits $\mathcal{O}(M)$ scaling.

Estimated memory requirements

The memory requirements of the method in our example calculations are classified as finite element matrices or particle-related arrays. As the number of elements increases, the finite element matrices comprise a majority of the total allocated memory, as demonstrated in Table 3.2 for the case of $N = 10^6$. As the polynomial order increases — e.g. $q = 4$, which corresponds with a 6th order basis for the finite element

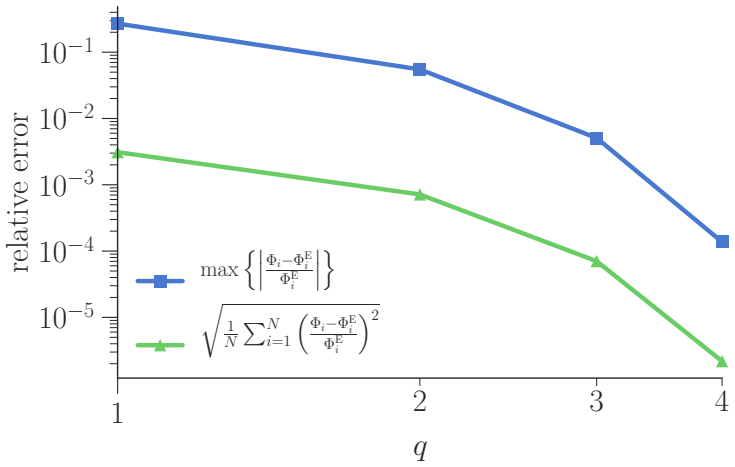


Figure 3.8: Convergence of the max and RMS measure of relative error in Φ versus q for a sample case with $N = 10\,000$.

Order of screens	Particle arrays	FE matrices ($N_{\mathrm{el}} = 9 \times 9 \times 9$)	FE matrices ($N_{\mathrm{el}} = 25 \times 25 \times 25$)
linear ($q = 1$):	184 MB	39 MB ($\approx 18\%$)	844 MB ($\approx 81\%$)
quartic ($q = 4$):	1120 MB	1290 MB ($\approx 52\%$)	27 600 MB ($\approx 95\%$)

Table 3.2: Memory requirements versus order of screens for $N = 10^6$.

solve — this effect increases as expected. We note that this memory footprint is typical for high-order FEM, but more optimal methods do exist [52].

Extension to Dirichlet boundary conditions

We now demonstrate the modification for Dirichlet boundary conditions described in Section 3.1. For these tests, we consider the unit cube with particles uniformly distributed within the smaller cube defined by $[0.1, 0.9]^3$. The particles are 55% positive and 45% negative; a net charge is allowed with Dirichlet boundary conditions, unlike the periodic problem. We first consider homogeneous boundary conditions on all faces of the domain. In this case, an exact solution is defined by the method of images. In order to compute this result, we turn the Dirichlet problem in $[0, 1]^3$ into a periodic problem in $[-1, 1]^3$. This periodic problem has eight times as many particles as the original problem, with seven sets of image charges chosen to have the proper charge strengths and reflected coordinates required for the method of images. We then solve with a high-resolution Ewald solver as before. The errors for the homogeneous Dirichlet case are shown in Table 3.3 for $N = 10\,000$. In addition, the values for the maximum relative errors in the $N = 10\,000$ case from the previous periodic test (Section 3.3 and Figure 3.8) are shown for comparison. We see similar errors for both

test cases.

Maximum relative error in FE-based method		
q	Homogeneous Dirichlet (Section 3.3)	Periodic (Section 3.3)
1	9.579×10^{-2}	2.701×10^{-1}
2	1.927×10^{-2}	5.507×10^{-2}
3	2.983×10^{-3}	5.077×10^{-3}
4	1.917×10^{-4}	1.464×10^{-4}

Table 3.3: Errors for the homogeneous Dirichlet and periodic test cases with $N = 10\,000$, $h = 0.06\bar{6}$, and screen order ranges from $q = 1$ to $q = 4$.

Next, we consider a non-homogeneous Dirichlet boundary condition, where $\Phi = g$ on ∂V and g is

$$g(\mathbf{x}) = \begin{cases} 2500\,[1 + 0.9\sin(2\pi y)\sin(\pi z)] & \text{when x} = 0 \\ 2500 & \text{otherwise.} \end{cases}$$

The particle distribution is the same as the homogeneous Dirichlet case (schematic of this case is depicted in Figure 3.9, with the gray box indicating the region where the particles are placed). The method of images alone can no longer solve this problem. Any method (Ewald-based, FMM, etc.) that relies on symmetry and images in order to implement different boundary conditions would not be able to complete this test on its own, but would need to be coupled with a Laplace solver capable of handling general boundary conditions. Our method, in contrast, only requires changing the description of the boundary condition from the previous homogeneous case.

A slice of the resulting potential is shown on the right side of Figure 3.9 and we note the expected behavior. For example, the overall potential is positive, with the peak value near the center of the box. Moreover, the contours flatten in the particle-free zone around the edge of the domain, as there are no longer large fluctuations from very near particles. The gradient of the potential (the electric field) becomes perpendicular to the walls in the center of each of the three constant-valued edges, as expected along conducting surfaces. Finally, the sinusoidal boundary condition "pulls" the contour lines along the $x = 0$ face of the domain.

3.4 Discussion

Comparison to PME

While the method proposed here incorporates several advantageous features of PME, there are notable differences that may offer benefits in certain settings. Our method no longer relies on the FFT, which may

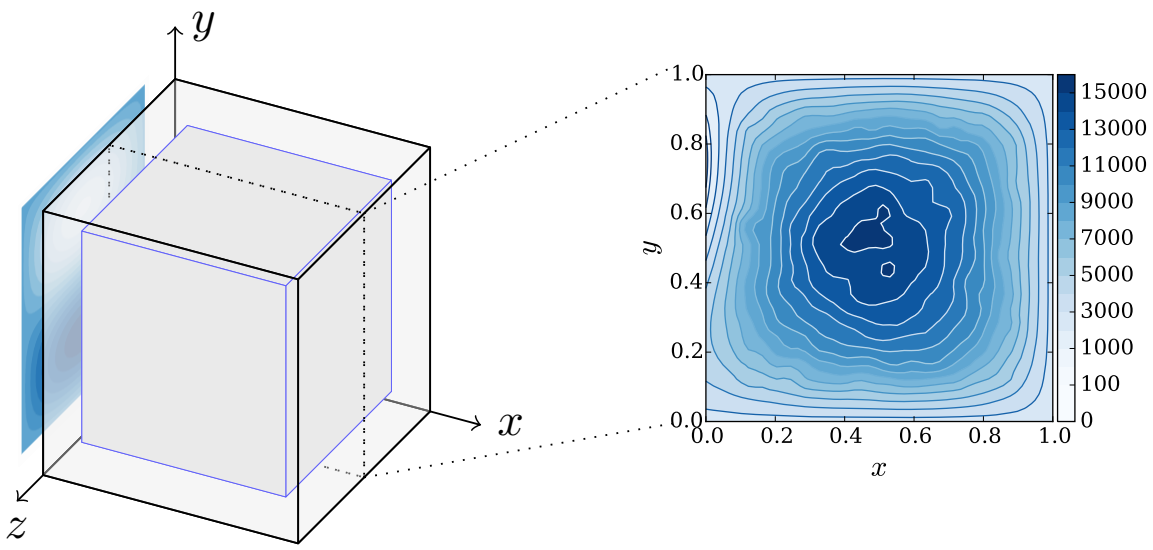


Figure 3.9: Schematic for non-homogeneous Dirichlet test case. The charges are distributed within the gray box; the non-constant boundary condition is shown at $x = 0$. A slice of the potential taken at $z = 0.5$ is shown on the right.

be limiting a extreme scales (in comparison to other Poisson solvers) and forces an assumption of structure on the compute geometry. The key is the introduction of a mesh-based screen, which introduces additional complexities locally, but also allows for a more general decomposition of the problem. There are particle-mesh variants that use finite elements — e.g., some PIC methods [53, 54] — but these have been proposed with a symmetric screen, which must be resolved on the mesh. We avoid this approximation, but at the cost of more intricate screen functions, which are constructed with (and the resulting potentials evaluated by) using memory-local operations.

These new screens incur an $\mathcal{O}(p^6)$ cost when represented by p-order basis functions as discussed in Section 3.2. This is larger than the $\mathcal{O}(\tilde{p}^3)$ cost of the $\tilde{p}$-order B-spline interpolations in PME. However, the polynomial order p in the present scheme and the B-spline order $\tilde{p}$ in PME are only loosely related. The B-spline order affects the overall accuracy of the PME method since it affects the resolution of the mesh description of the smooth potential. The polynomial order p in the present method does not have the same effect, since the screens are exactly represented on the mesh. Instead, p affects $\hat{R}_c$ via the decay of the screened potential as shown in Figure 3.3. This is important, since for uniform charge density the cost of point-to-point interactions scales with volume $\sim \hat{R}_c^3$. An independent Ewald splitting parameter sets the corresponding truncation error at fixed cut-off radius for PME.

Similarly, the mesh density has different implications in the two methods. As with the B-spline order, the mesh density in PME affects the accuracy by providing more resolution for the potential. A denser mesh

does not affect the short-range calculation, but requires more global communication for the FFT. In contrast, the mesh density in the present scheme decreases the communication burden for the short-range component of the calculation by reducing the number of interactions included for a given $\hat{R}_c$, since the cut-off radius is scaled by the mesh size, unlike in PME. The communication required of the mesh solver is that of multigrid.

The simple periodic test case depicted in Section 3.3 allows computation of the "exact" answer using the Ewald sum in order to report errors. A triply-periodic box, however, is an ideal case for PME, and we would expect its performance in terms of total CPU time to be superior to other methods, including the method presented here. This is confirmed for the case with $N = 10\,000$ particles that is shown in Figure 3.8, with timing results summarized in Table 3.4. Here, p is the order of interpolation for a standard PME method with $\alpha = 100$ and cutoff radius $R_c = 0.23$. This was selected to approximately equal the size used in our FE-based method. The number of mesh points for the PME method was $M = 97,336$, which corresponds to the number of degrees of freedom in the mesh for the FE-based case with $q = 1$.

Method		Max relative abs. error	Short-range/ self time (sec)	Mesh solve time (sec)
FE-based screens	$q = 1$	2.70×10^{-1}	7.58×10^{0}	1.06×10^{1}
	$q = 2$	5.51×10^{-2}	2.51×10^{1}	4.48×10^{1}
	$q = 3$	5.08×10^{-3}	7.22×10^{1}	6.49×10^{1}
	$q = 4$	1.46×10^{-4}	2.17×10^{2}	1.66×10^{2}
PME	$p = 1$	5.47×10^{-1}	1.12×10^{0}	6.00×10^{-3}
	$p = 2$	2.35×10^{-2}	1.53×10^{0}	8.00×10^{-3}
	$p = 3$	6.47×10^{-3}	1.21×10^{0}	6.00×10^{-3}

Table 3.4: CPU times for major components of a periodic potential calculation.

The PME short-range/self time is nearly the same for all values of p, since the short-range term in PME does not depend on the order of interpolation. In both PME and our FE-based method, we expect the short-range time to be drastically reduced when the problem size is large enough to allow a high degree of parallelization. Furthermore, both methods are significantly faster than the standard Ewald sum for this problem, which took $\sim 3 \times 10^3$ seconds.

Comparison to FMM

The method presented in this paper shares several attractive features of the fast multipole method, most notably the linear scaling. The relative merits in comparison to FMM are likely application dependent, and the preferred choice depends on several factors. Though intricate, the low communication burden of FMM leads to efficient implementations [55]. Both methods become expensive with increased p, the basis order

in the present scheme or the multipole expansion order for FMM. Yet, the highly local work load of the proposed high-order screens is more suitable for emerging architectures with accelerators.

Chapter 4

High-order finite element–integral equation coupling

We now turn our attention to coupling Green's functions and finite elements through layer potential integral equations for embedded mesh problems, with a solution decomposition in the spirit of [19]. The method results in strong enforcement of essential boundary conditions on the embedded domain and does not require any modification of the finite element weak form integral operators or basis functions. We show how to solve both interior and exterior embedded domain problems, with a new FE-IE splitting introduced for the latter.

4.1 Background and notation

Assume a smooth domain $\Omega \subset \mathbb{R}^d$ with $d = 2, 3$. For a linear partial differential operator $\mathcal{L}$ with associated free-space Green's function $G(x, x_0)$, we define the single- and double-layer potential operators on a density function γ as, respectively,

$$\mathcal{S}\gamma(x) = \int_{\partial\Omega} G(x, x_0)\gamma(x_0)\, dx_0 \tag{4.1}$$

and

$$\mathcal{D}\gamma(x) = \int_{\partial\Omega} \left(\nabla_{x_0} G(x, x_0) \cdot \mathbf{n}(x_0)\right) \gamma(x_0)\, dx_0. \tag{4.2}$$

Here ∇_{x_0} denotes the gradient with respect to the variable of integration and $\mathbf{n}(x_0)$ is the outward-facing normal vector. The single-layer potential is thus the limit of a collection of point forces along the boundary with strengths given by γ and the double-layer potential is a similar collection of point dipoles. We also define the normal derivatives of the layer potentials,

$$\mathcal{S}'\gamma(x) = \int_{\partial\Omega} \left(\nabla_{x} G(x, x_0) \cdot \mathbf{n}(x)\right) \gamma(x_0)\, dx_0 \tag{4.3}$$

and

$$\mathcal{D}'\gamma(x) = \int_{\partial\Omega} \left(\nabla_{x} \left(\nabla_{x_0} G(x, x_0) \cdot \mathbf{n}(x_0)\right) \cdot \mathbf{n}(x)\right) \gamma(x_0)\, dx_0. \tag{4.4}$$

Because the Green's function solves $\mathcal{L}G(x, x_0) = 0$ everywhere away from the boundary, a layer potential

representation $u = \mathcal{S}\gamma$ or $u = \mathcal{D}\gamma$ will satisfy the homogeneous governing equation $\mathcal{L}u = 0$. For the Laplace operator, the Green's functions in two and three dimensions are

$$G(x, x_0) = \begin{cases} -\frac{1}{2\pi} \log |x - x_0| & d = 2, \\ \frac{1}{4\pi} \frac{1}{|x - x_0|} & d = 3. \end{cases} \tag{4.5}$$

To solve boundary value problems we must relate the boundary behavior of the layer potentials to our boundary conditions. For domains with smooth boundaries we have the following relationships between the interior limits (subscript $-$), the exterior limits (subscript $+$), and the on-boundary values (enclosed in parentheses):

$$\begin{aligned} \mathcal{S}'_- \gamma = & \quad \frac{1}{2}\gamma + (\mathcal{S}')\gamma \quad \text{and} \\ \mathcal{S}'_+ \gamma = & -\frac{1}{2}\gamma + (\mathcal{S}')\gamma \end{aligned} \tag{4.6}$$

for the normal derivative of the single-layer potential, and

$$\begin{aligned} \mathcal{D}_- \gamma = & -\frac{1}{2}\gamma + (\mathcal{D})\gamma \quad \text{and} \\ \mathcal{D}_+ \gamma = & \quad \frac{1}{2}\gamma + (\mathcal{D})\gamma \end{aligned} \tag{4.7}$$

for the double-layer potential. The on-boundary values are defined as the Cauchy principal value of the integrals. The single-layer potential and the normal derivative of the double-layer potential are continuous at the boundary, or

$$\mathcal{S}_- \gamma = \mathcal{S}_+ \gamma = (\mathcal{S})\gamma \tag{4.8}$$

and

$$\mathcal{D}'_- \gamma = \mathcal{D}'_+ \gamma = (\mathcal{D}')\gamma. \tag{4.9}$$

The $\mathcal{D}'$ operator must be defined as the finite part of a hypersingular integral.

Consider a general Fredholm integral equation

$$\{cI + (\mathcal{A})\}\, \gamma = g \tag{4.10}$$

on the boundary $\partial\Omega$, where c is a constant, I is the identity, and $\mathcal{A}$ is an integral operator containing some combination of $\mathcal{S}$, $\mathcal{S}'$, $\mathcal{D}$, or $\mathcal{D}'$. We assume γ, $g \in X$ where X is a suitable Banach space for which

$(\mathcal{A}) : X \to X$ is a compact operator. When $c = 0$, (4.10) is an equation of the first kind, which is well-known to be ill-posed and lead to ill-conditioned systems of equations in the discrete problem. When $c \neq 0$, we have a second kind equation, which leads to well-conditioned system of equations. Furthermore, if $cI + (\mathcal{A})$ has no nullspace, we are guaranteed existence and uniqueness of the solution γ for any $g \in X$ by the Fredholm alternative [1]. Thus our goal in choosing a layer potential representation will be that its on-boundary behavior (4.6-4.9) results in a second-kind equation for the desired boundary conditions.

4.2 Interior problems

We first consider the combined finite element–integral equation (FE-IE) method introduced by Rüberg and Cirak [19]. We present an error analysis for our implementation of the method and demonstrate the convergence for problems of varying smoothness.

Assume a smooth domain $\Omega \subset \mathbb{R}^n$. Consider the boundary value problem

$$\begin{aligned} -\nabla \cdot \nabla u(x) &= f \qquad x \in \Omega \\ u(x) &= g \qquad x \in \partial\Omega. \end{aligned} \tag{4.11}$$

We introduce a convex polygonal domain $\hat{\Omega}$ such that $\Omega \subset \hat{\Omega}$. Equation (4.11) is split into two problems such that $u = u_1 + u_2$, with u_1 as the finite element solution solved on $\hat{\Omega}$ and u_2 the integral equation solution defined in Ω. We define r_i as the restriction of u_i to the "true" domain boundary $\partial\Omega$ and $\hat{r}_i$ as the restriction of u_i to the fictitious boundary $\partial\hat{\Omega}$. We can then write our two problems as

$$\begin{aligned} \text{[FE]} \quad -\nabla \cdot \nabla u_1(x) &= f \qquad x \in \hat{\Omega} \\ \hat{r}_1 &= 0 \end{aligned} \qquad\qquad \begin{aligned} \text{[IE]} \quad -\nabla \cdot \nabla u_2(x) &= 0 \qquad x \in \Omega \\ r_2 &= g - r_1. \end{aligned} \tag{4.12}$$

We note that because u_1 does not depend on u_2, we can solve the two problems separately and sequentially. Furthermore, we see that the integral equation solution u_2 solves the Laplace equation; this way, we let the finite element solver handle the right hand side of the original problem (4.11) and avoid a costly volume integration in the solution of u_2.

Finite element formulation

First, we solve the uncoupled FE problem. The Poisson equation (4.12) has the following associated weak equation: *Find* $u_1 \in H_0^1(\hat{\Omega})$ *such that*

$$\int_{\hat{\Omega}} \nabla u_1 \cdot \nabla v \, dV = \int_{\hat{\Omega}} f v \, dV \quad \forall v \in H_0^1(\hat{\Omega}), \tag{4.13}$$

where $H_0^1(\hat{\Omega})$ indicates functions in H^1 with compact support in the fictitious domain $\hat{\Omega}$.

We introduce the operator $\mathcal{F}(v)$, which represents the application of the bilinear form for the finite element problem, i.e.,

$$\mathcal{F}(v)\, u_1 = \int_{\hat{\Omega}} \nabla u_1 \cdot \nabla v \, dV \tag{4.14}$$

for $v \in H_0^1(\hat{\Omega})$. We also define $\mathcal{M}(v)$ as the linear functional operator

$$\mathcal{M}(v)\, f = \int_{\hat{\Omega}} f v \, dV$$

for $v \in H_0^1(\hat{\Omega})$. Then the finite element problem in (4.12) is: *Find* $u_1 \in H_0^1(\hat{\Omega})$ *such that*

$$\mathcal{F}(v)\, u_1 = \mathcal{M}(v)\, f \quad \forall v \in H_0^1(\hat{\Omega}). \tag{4.15}$$

Layer potential representation

Next we solve the IE portion of (4.12). The IE solution u_2 corrects for the effects of the FE solution on the boundary. As previously mentioned, we seek a layer potential representation that results in a second-kind equation. In this case, we are solving an interior Dirichlet problem. Let $u_2 = \mathcal{D}\gamma$, the double-layer potential. Then after applying the jump condition (4.7), the integral equation problem in (4.12) is

$$\left\{ -\frac{1}{2} I + (\mathcal{D}) \right\} \gamma = g - r_1. \tag{4.16}$$

This operator is known to have a trivial nullspace for the Laplace Green's function [1] and thus we are guaranteed existence and uniqueness of the equation by the Fredholm alternative.

Error convergence

Now that we have defined our individual FE and IE boundary value problems, we might naturally want to analyze what changes, if any, occur in the coupled solution's error convergence when compared with the

standard convergence theory of the individual methods. We will now show the components of the error for the combined solution and their convergence depending on the smoothness of the problem.

Due to the linearity of the solution, the total error in the solution e_u satisfies

$$\|e_u\| \leq \|e_{u_1}\| + \|e_{u_2}\|. \tag{4.17}$$

We begin our discussion with the IE solution, $u_2 = \mathcal{A}\gamma$, where $\mathcal{A} : X \to Y$ is a compact linear operator from Banach space X to Banach space Y. We assume that $(\mathcal{A}) : X \to X$, the on-boundary evaluation of the integral representation, is also compact. Furthermore, we assume that we have a collectively compact set of discrete approximations to $\mathcal{A}$, with a discrete representation denoted $\mathcal{A}^h$.

Remark 4.1. *Though we chose $\mathcal{A} = \mathcal{D}$ for the Dirichlet problem above, we leave the operator more general here, as the analysis applies in the case of other representations as well. The only restriction on $\mathcal{A}$ is that the resulting integral equation to be solved is second kind and that the resulting operator $\{cI + (\mathcal{A})\}$ has a trivial nullspace.*

The error in the numerical solution u_2^h comes from two sources: approximation of $\mathcal{A}$ and error in γ. To this end, we define the $E_{\mathcal{A}} \equiv \mathcal{A}^h - \mathcal{A}$ as the error in the approximated representation operator $\mathcal{A}^h$. That is, the exact solution u_2 satisfies

$$u_2 = \mathcal{A}\gamma = \mathcal{A}^h\gamma - E_{\mathcal{A}}\gamma. \tag{4.18}$$

We also define the approximate density γ^h such that the error in γ is $e_\gamma \equiv \gamma^h - \gamma$. Thus the error $e_{u_2} \equiv u_2^h - u_2$ is

$$u_2^h - u_2 = \mathcal{A}^h(\gamma^h - \gamma) + E_{\mathcal{A}}\gamma,$$

or

$$e_{u_2} = \mathcal{A}^h e_\gamma + E_{\mathcal{A}}\gamma.$$

Because $\mathcal{A}$ is compact and thus bounded, we can write

$$\|e_{u_2}\|_Y \leq C(\|e_\gamma\|_X + \|E_{\mathcal{A}}\gamma\|_Y) \tag{4.19}$$

for some constant C. Consequently, we also consider the *components* of the error e_γ. First we examine the error due to the representation of $\mathcal{A}$. For a Dirichlet problem with boundary condition given by b, the exact density γ solves

$$\{cI + (\mathcal{A})\}\,\gamma = b, \tag{4.20}$$

for some constant c, where we recall that the notation $(\mathcal{A})$ denotes the evaluation of the boundary integral operator on the boundary itself. The operator $(\mathcal{A}) : X \to X$ for the Banach space X is assumed to be compact. Next, we introduce $\tilde{\gamma}^h$, an approximate solution, for which

$$\left\{cI + (\mathcal{A})^h\right\}\tilde{\gamma}^h = b, \tag{4.21}$$

and expand (4.20) as

$$\left\{cI + (\mathcal{A})^h\right\}\gamma - (E_{\mathcal{A}})\gamma = b. \tag{4.22}$$

Applying $\left\{cI + (\mathcal{A})^h\right\}^{-1}$ to (4.21) and (4.22) we have

$$\tilde{\gamma}^h = \left\{cI + (\mathcal{A})^h\right\}^{-1} b, \quad \text{and} \tag{4.23}$$

$$\gamma \; - \left[\left\{cI + (\mathcal{A})^h\right\}^{-1} (E_{\mathcal{A}})\right]\gamma = \left\{cI + (\mathcal{A})^h\right\}^{-1} b. \tag{4.24}$$

The right hand sides of (4.23) and (4.24) are identical, which leads to

$$\tilde{\gamma}^h = \gamma - \left[\left\{cI + (\mathcal{A})^h\right\}^{-1} (E_{\mathcal{A}})\right]\gamma. \tag{4.25}$$

In addition, we define the error in γ^h due to the approximation of $\mathcal{A}$, $e_{\sigma,\mathcal{A}} \equiv \tilde{\gamma}^h - \gamma$, and see that

$$e_{\sigma,\mathcal{A}} = -\left[\left\{cI + (\mathcal{A})^h\right\}^{-1} (E_{\mathcal{A}})\right]\gamma. \tag{4.26}$$

In the FE-IE method, our boundary condition b contains a correction for the finite element solution on $\partial\Omega$, so that $b = g - r_1$. Thus the FE solution also introduces error in γ due to error in the boundary condition, which we define as $e_{\sigma,\mathcal{B}} \equiv \gamma^h - \tilde{\gamma}^h$. If we consider the exact FE solution restriction as r_1, then we rewrite (4.21) as

$$\left\{cI + (\mathcal{A})^h\right\}\tilde{\gamma}^h = g - r_1.$$

In contrast, the actual approximated density γ^h solves

$$\left\{cI + (\mathcal{A})^h\right\}\gamma^h = g - r_1^h.$$

Then, combining these equations we see that

$$\left\{cI + (\mathcal{A})^h\right\}(\gamma^h - \tilde{\gamma}^h) = -r_1^h + r_1,$$

or

$$\left\{cI + (\mathcal{A})^h\right\} e_{\sigma,\mathcal{B}} = -e_{r_1}. \tag{4.27}$$

Thus the error $e_{\sigma,\mathcal{B}}$ is

$$e_{\sigma,\mathcal{B}} = -\left\{cI + (\mathcal{A})^h\right\}^{-1} e_{r_1}. \tag{4.28}$$

Finally we see that $\gamma^h = e_{\sigma,\mathcal{B}} + \tilde{\gamma}^h = e_{\sigma,\mathcal{B}} + e_{\sigma,\mathcal{A}} + \gamma$, or

$$e_\gamma = e_{\sigma,\mathcal{B}} + e_{\sigma,\mathcal{A}} = -\left[\left\{cI + (\mathcal{A})^h\right\}^{-1} (E_{\mathcal{A}})\right] \gamma - \left\{cI + (\mathcal{A})^h\right\}^{-1} e_{r_1}. \tag{4.29}$$

Moreover, because $\left\{cI + (\mathcal{A})^h\right\}$ has a bounded inverse [1], the norm of e_γ is bounded by

$$\|e_\gamma\|_X \leq C\left(\|(E_{\mathcal{A}})\gamma\|_X + \|e_{r_1}\|_X\right). \tag{4.30}$$

Now we return to our error bound for u_2, given in (4.19), and see that

$$\|e_{u_2}\|_Y \leq C\left(\|E_{\mathcal{A}}\gamma\|_Y + \|(E_{\mathcal{A}})\gamma\|_X + \|e_{r_1}\|_X\right), \tag{4.31}$$

for some constant C. The error in r_1 is dictated by the convergence of the finite element solution u_1, so we appeal to the usual and well-developed finite element error analysis. Of course this also holds for e_{u_1}, so we return to the total error:

$$\|e_u\|_Y \leq C\left(\|e_{u_1}\|_Y + \|E_{\mathcal{A}}\gamma\|_Y + \|(E_{\mathcal{A}})\gamma\|_X + \|e_{r_1}\|_X\right). \tag{4.32}$$

Next, we select X and Y. For example, if $X = C(\partial\Omega)$, the space of continuous functions on the boundary equipped with the maximum norm, then $\|\cdot\|_X = \|\cdot\|_{L^\infty(\partial\Omega)}$. Since $\|\cdot\|_{L^\infty(\partial\Omega)} \leq \|\cdot\|_{L^\infty(\Omega)}$, (4.31) becomes

$$\|e_{u_2}\|_Y \leq C\left(\|E_{\mathcal{A}}\gamma\|_Y + \|(E_{\mathcal{A}})\gamma\|_X + \|e_{u_1}\|_{L^\infty(\Omega)}\right). \tag{4.33}$$

Furthermore, if we assume a continuous solution and choose $\|\cdot\|_Y = \|\cdot\|_{L^\infty(\Omega)}$, then the total error is

$$\|e_u\|_{L^\infty(\Omega)} \leq C\left(\|e_{u_1}\|_{L^\infty(\Omega)} + \|E_{\mathcal{A}}\gamma\|_{L^\infty(\Omega)} + \|(E_{\mathcal{A}})\gamma\|_{L^\infty(\partial\Omega)} + \|e_{u_1}\|_{L^\infty(\Omega)}\right).$$

Collecting the e_{u_1} terms yields

$$\|e_u\|_{L^\infty(\Omega)} \leq C \left(\|e_{u_1}\|_{L^\infty(\Omega)} + \|(E_{\mathcal{A}})\gamma\|_{L^\infty(\partial\Omega)} + \|E_{\mathcal{A}}\gamma\|_{L^\infty(\Omega)} \right), \tag{4.34}$$

indicating that the maximum error in the total solution is bounded separately by the finite element error and integral operator error.

Alternatively, weaker restrictions on X and Y, such as $X = H^{1/2}(\partial\Omega)$ and $Y = H^1(\Omega)$, result in

$$\|e_u\|_{H^1(\Omega)} \leq C \left(\|e_{u_1}\|_{H^1(\Omega)} + \|E_{\mathcal{A}}\gamma\|_{H^1(\Omega)} + \|(E_{\mathcal{A}})\gamma\|_{H^{1/2}(\partial\Omega)} + \|e_{r_1}\|_{H^{1/2}(\partial\Omega)} \right). \tag{4.35}$$

Or, as $\|\cdot\|_{L^2(\Omega)} \leq \|\cdot\|_{H^1(\Omega)}$,

$$\|e_u\|_{L^2(\Omega)} \leq C \left(\|e_{u_1}\|_{H^1(\Omega)} + \|E_{\mathcal{A}}\gamma\|_{H^1(\Omega)} + \|(E_{\mathcal{A}})\gamma\|_{H^{1/2}(\partial\Omega)} + \|e_{r_1}\|_{H^{1/2}(\partial\Omega)} \right). \tag{4.36}$$

In other words, our convergence is again bounded separately by the finite element and integral operator error, even with weaker restrictions on the smoothness of the problem.

Demonstration of convergence

The implementation of a combined FE-IE solver here improves on the numerical methods in [19], where a single-layer representation is used for u_2. This leads to an integral equation of the first kind, that must be regularized. In contrast, we use a double-layer representation, which gives an integral equation of the second kind and results in a well-conditioned numerical system. Though the single- and double-layer operators (4.1–4.2) are well-defined on the boundary (for the double-layer operator, this is understood in a principal value sense), their kernels are singular when the evaluation point is on the boundary. In other words, for every point in the discretization of the boundary, the kernel of the layer potential operator includes a singularity, necessitating special treatment for accurate computation of the resulting system of linear equations. For each equation, a coordinate transformation is used in [19] to remove this singularity and adaptive quadrature is employed for points near the singularity. We use "quadrature by expansion" (QBX) [56] instead. The finite element calculations were carried out with the use of the `deal.II` library [57, 58]. Standard Q_p elements were used, where p is the order of the basis functions. The integral equation code uses `pytential` [59] and its dependencies.

It is simple to extend our implementation to higher orders; one simply has to increase the order of the individual components — i.e., increase the order of the finite element basis functions p and the order of

QBX used. The order of QBX was chosen to correspond to the expected optimal order of convergence of the finite element solution. Furthermore, the method is high-order *even very near to the embedded boundary*, in contrast with standard immersed boundary methods. This is demonstrated by the results summarized in Table 4.1, where self-convergence tests are shown for three different right hand side functions $f(x)$. For our Poisson problem, we are only guaranteed the existence of weak derivatives two orders higher than what exists in the right hand side. Table 4.1 compares three different right hand sides: constant function

$$f_c(x, y) = 1, \tag{4.37}$$

which admits a classical strong solution in C^2; piecewise continuous bilinear function

$$f_{\text{bl}}(x, y) = \xi(x)\xi(y), \qquad \text{where} \qquad \xi(z) = \begin{cases} \frac{5}{3}z + 1 & \text{if} \quad z \leq 0 \\ -\frac{5}{3}z + 1 & \text{if} \quad z > 0, \end{cases} \tag{4.38}$$

which is in H^1; and piecewise constant function

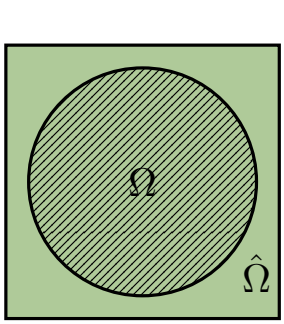
$$f_{\text{pw}} = \eta(x)\eta(y), \qquad \text{where} \qquad \eta(z) = \begin{cases} -1 & \text{if} \quad z \leq 0 \\ 1 & \text{if} \quad z > 0, \end{cases} \tag{4.39}$$

which is only in L^2, but not in H^1.

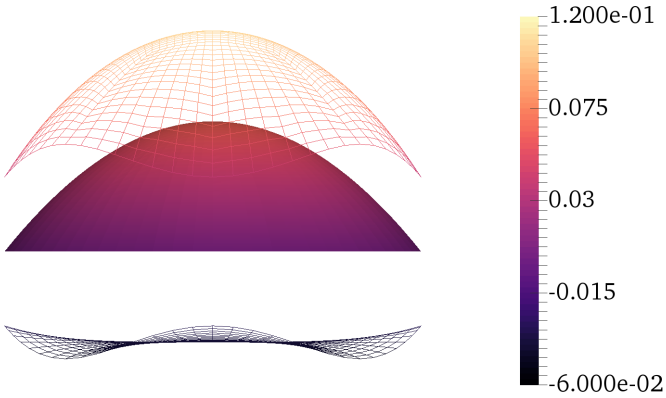
All of the problems in Table 4.1 are defined on the domain $\Omega = \{x : |x| \leq 0.5\}$, the circle of radius 0.5. The domain is embedded in a square domain $\hat{\Omega} = [-0.6, -0.6] \times [0.6, 0.6]$. This is illustrated, along with the solution for the constant right hand side, in Figure 4.1.

We see that the method retains high order as allowed by the smoothness of the problem. We note the difference in behavior for $\|\cdot\|_{L^\infty(\Omega)}$ convergence and $\|\cdot\|_{L^2(\Omega)}$ convergence at higher orders in the two piecewise $f(x)$ cases. This is expected as the $\|\cdot\|_{L^\infty(\Omega)}$ behavior for QBX is defined in terms of strong derivatives, which is a stricter condition on smoothness [60]. Also, for linear basis functions, the finite element convergence rate in $\|\cdot\|_{L^\infty}$ is not quite optimal, as it contains an extra factor of $\log(1/h)$ (cf [61, 62]).

Remark 4.2. *The mesh for the IE solver in 2D is a parametric curve. As such, especially in the case of domains like the ones described in (4.51) and (4.52), the rate at which the maximum mesh spacing h decreases with increasing number of elements will not always be the same as for the FE solver. In this case, we select whichever change in h results in the more conservative estimate for the empirical order of convergence.*



(a) FE domain $\hat{\Omega} = [-0.6, -0.6] \times [0.6, 0.6]$ surrounds "true" domain Ω, where u_2 is defined.



(b) FE solution (top wireframe), IE solution (bottom wireframe), and total solution (solid) with right hand side (4.37).

Figure 4.1: Solution domains and sample solution for an interior embedded mesh calculation.

As we see the effect that the smoothness of the right hand side has on the convergence of the method, an important point to consider is the case when data for f is only available on Ω. If it is necessary to compute an extension to f in $\hat{\Omega}\backslash\Omega$, the smoothness of the resulting f may determine the overall convergence rate of the method.

4.3 Exterior problems

In this section we extend the method to domains with exclusions. Consider the *complement* of the previous interior Poisson problem, where the solution is sought outside Ω but inside a polygon $\hat{\Omega}$:

$$\begin{aligned} -\nabla \cdot \nabla u(x) &= f(x) \qquad x \in \hat{\Omega}\backslash\Omega \\ u(x) &= g(x) \qquad x \in \partial\Omega \\ u(x) &= \hat{g}(x) \qquad x \in \partial\hat{\Omega}. \end{aligned} \tag{4.40}$$

Our new approach for the FE-IE decomposition of this problem is to solve an interior finite problem and an exterior integral equation problem, with the two solutions coupled only through their boundary values. These domains are illustrated in Figure 4.2. In this way, we let the two methods play to their individual strengths: the finite element solution exists on a regular, rectangular mesh with no exclusions and handles the forcing term f, while the layer potential solution takes care of the interior boundary condition on $\partial\Omega$

definition			*convergence*				
rhs	p	qbx	h_{fe}, h_{ie}	$\|\text{diff}\|_{L^\infty(\Omega)}$	order	$\|\text{diff}\|_{L^2(\Omega)}$	order
constant, eq. (4.37)	1	2	0.04, 0.105	4.564×10^{-4}	–	9.576×10^{-5}	–
			0.02, 0.052	9.333×10^{-5}	2.3	1.975×10^{-5}	2.3
			0.01, 0.026	1.812×10^{-5}	2.4	3.982×10^{-6}	2.3
	2	3	0.04, 0.105	9.383×10^{-5}	–	2.189×10^{-5}	–
			0.02, 0.052	1.032×10^{-5}	3.2	2.414×10^{-6}	3.2
			0.01, 0.026	8.840×10^{-7}	3.5	2.052×10^{-7}	3.6
	3	4	0.04, 0.105	2.405×10^{-5}	–	6.072×10^{-6}	–
			0.02, 0.052	1.510×10^{-6}	4.9	3.727×10^{-7}	4.0
			0.01, 0.026	6.887×10^{-8}	4.5	1.672×10^{-8}	4.5
piecewise continuous bilinear, eq. (4.38)	1	2	0.04, 0.105	1.091×10^{-4}	–	3.040×10^{-5}	–
			0.02, 0.052	2.110×10^{-5}	2.4	6.377×10^{-6}	2.3
			0.01, 0.026	4.201×10^{-6}	2.3	1.555×10^{-6}	2.0
	2	3	0.04, 0.105	2.294×10^{-5}	–	5.736×10^{-6}	–
			0.02, 0.052	2.453×10^{-6}	3.2	6.375×10^{-7}	3.2
			0.01, 0.026	2.219×10^{-7}	3.5	5.424×10^{-8}	3.6
	3	4	0.04, 0.105	5.594×10^{-6}	–	1.597×10^{-6}	–
			0.02, 0.052	4.103×10^{-7}	3.8	9.799×10^{-8}	4.0
			0.01, 0.026	2.396×10^{-8}	4.1	4.417×10^{-9}	4.5
piecewise constant, eq. (4.39)	1	2	0.04, 0.105	4.918×10^{-4}	–	1.008×10^{-4}	–
			0.02, 0.052	1.882×10^{-4}	1.4	2.475×10^{-5}	2.0
			0.01, 0.026	5.620×10^{-5}	1.7	4.222×10^{-6}	2.6
	2	3	0.04, 0.105	2.811×10^{-4}	–	2.417×10^{-5}	–
			0.02, 0.052	9.098×10^{-5}	1.6	3.583×10^{-6}	2.8
			0.01, 0.026	2.489×10^{-5}	1.9	4.847×10^{-7}	2.9
	3	4	0.04, 0.105	1.775×10^{-4}	–	9.453×10^{-6}	–
			0.02, 0.052	4.730×10^{-5}	1.9	1.294×10^{-6}	2.9
			0.01, 0.026	1.219×10^{-5}	2.0	1.594×10^{-7}	3.0
— *all* —	4	5	0.005, 0.013	— *reference solution parameters* —			

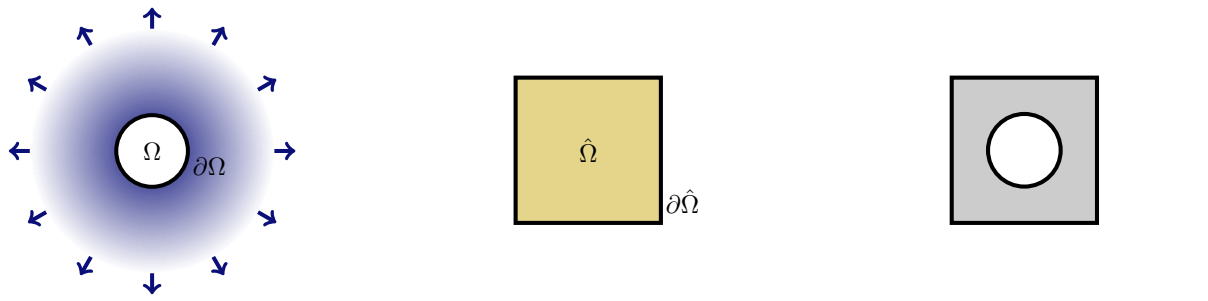
Table 4.1: Convergence to a fine mesh solution vs. smoothness of rhs for the interior problem.

but does not include the corners of $\partial\hat{\Omega}$, which would cause discontinuities in the density γ.

The FE and IE problems are now defined as

$$\begin{aligned} &\text{[FE]} \quad -\nabla\cdot\nabla u_1(x) = f(x) \qquad x\in\hat{\Omega} && \text{[IE]} \quad -\nabla\cdot\nabla u_2(x) = 0 \qquad x\in\mathbb{R}^d\backslash\Omega \\ &\qquad\qquad \hat{r}_1 = \hat{g}(x) - \hat{r}_2 \quad x\in\partial\hat{\Omega} && \qquad\qquad r_2 = g(x) - r_1 \quad x\in\partial\Omega. \end{aligned} \tag{4.41}$$

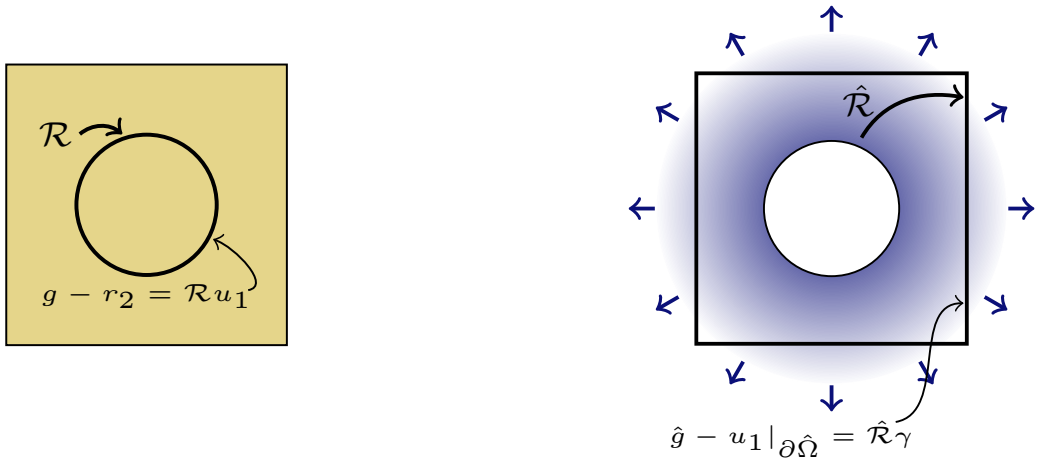
The problems are no longer able to be solved sequentially. The coupling through the boundary conditions



(a) IE domain $\mathbb{R}^d \backslash \Omega$ (b) FE domain $\hat{\Omega}$ (c) Original problem domain $\hat{\Omega} \backslash \Omega$

Figure 4.2: Three domains for a coupled FE-IE problem on a domain with a circular hole.

can be handled two ways: solve iteratively between the two until convergence is reached, or combine both into a single, large linear system. Next, we discuss the details of forming this coupled system.



(a) The $\mathcal{R}$ operator takes the FE solution and returns its restriction to the embedded boundary $\partial\Omega$.

(b) The $\hat{\mathcal{R}}$ operator takes a density γ and returns the restriction of the layer potential representation $\mathcal{A}\gamma$ at the boundary $\partial\hat{\Omega}$.

Figure 4.3: Coupling operators R and $\hat{R}$.

First, we define the operator $\mathcal{R}$ as the restriction to the "true" domain boundary $\partial\Omega$ and r_i as $\mathcal{R}u_i$. Likewise, define $\hat{\mathcal{R}}$ as the operator that takes a density γ on the boundary $\partial\Omega$ and returns the restriction of the layer potential representation $\mathcal{A}\gamma$ to the fictitious boundary $\partial\hat{\Omega}$. This is illustrated in Figure 4.3.

We define u_2 in terms of a layer potential operator $\mathcal{A}$ such that $u_2 = \mathcal{A}\gamma$ and the resulting integral equation is second kind:

$$\{cI + (\mathcal{A})\}\,\gamma = g - \mathcal{R}u_1, \tag{4.42}$$

for some constant c. Since u_1 is also unknown, this results in

$$\{cI + (\mathcal{A})\}\,\gamma + \mathcal{R}u_1 = g. \tag{4.43}$$

Let $\hat{\mathcal{F}}(v)$ be the FE bilinear form operator with modifications for the essential boundary conditions and $\hat{\mathcal{M}}(v;\hat{g})$ as the modified version of $\mathcal{M}(v)$ for boundary conditions $\hat{g}$:

$$\hat{\mathcal{F}}(v)\,u_1 = \begin{cases} u_1 & x \in \partial\hat{\Omega} \\ \mathcal{F}(v)\,u_1 & \text{otherwise} \end{cases} \quad \text{and} \quad \hat{\mathcal{M}}(v;\hat{g})\,f = \begin{cases} \hat{g} & x \in \partial\hat{\Omega} \\ \mathcal{M}(v)\,f & \text{otherwise.} \end{cases} \tag{4.44}$$

The weak form from (4.15) with boundary condition $\hat{g}$ is now written as: *Find* $u_1 \in H^1(\hat{\Omega})$ *such that*

$$\hat{\mathcal{F}}(v)\,u_1 = \hat{\mathcal{M}}(v;\hat{g})f \quad \forall v \in H_0^1(\hat{\Omega}). \tag{4.45}$$

As a result, the coupled problem to be solved is: *Find* γ, u_1 *such that*

$$\left[\begin{array}{c|c} cI + (\mathcal{A}) & \mathcal{R} \\ \hline \hat{\mathcal{R}} & \hat{\mathcal{F}}(v) \end{array}\right] \left[\begin{array}{c} \gamma \\ \hline u_1 \end{array}\right] = \left[\begin{array}{c} g \\ \hline \hat{\mathcal{M}}(v;\hat{g})f \end{array}\right] \qquad \forall v \in H_0^1(\hat{\Omega}). \tag{4.46}$$

Existence and uniqueness of the decomposition

To study the key aspects of (4.46), we introduce the "split solution" decomposition — i.e., taking a Schur complement approach. We define $\hat{\mathcal{F}}^{-1}$ as the operator such that $u = \hat{\mathcal{F}}^{-1}\hat{\mathcal{M}}(v;\hat{g})f$ solves the problem (4.45). The solution proceeds as follows:

1. Note that $u_1 = \hat{\mathcal{F}}^{-1}\hat{\mathcal{M}}(v;\hat{g})f - \hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\gamma$, and

2. Solve

$$\left\{cI + (\mathcal{A}) - \mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\right\}\gamma = g - \mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{M}}(v;\hat{g})\,f, \tag{4.47}$$

   then

3. Solve

$$u_1 = \hat{\mathcal{F}}^{-1}\hat{\mathcal{M}}(v;\hat{g})f - \hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\gamma. \tag{4.48}$$

Remark 4.3. *We note that as* $\hat{\mathcal{R}}\gamma$ *returns a function only on* $\partial\hat{\Omega}$ *and nothing interior to* $\hat{\Omega}$*, the term* $\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\gamma$ *could also be written* $\hat{\mathcal{F}}^{-1}\hat{\mathcal{M}}(v;\hat{\mathcal{R}}\gamma)\,\mathbf{0}$*. In this form, we see immediately that it is the solution to the*

weak Laplace equation with boundary conditions equal to $\hat{\mathcal{R}}\gamma$.

We need to choose a representation $\mathcal{A}$ for u_2. In the exterior, the double layer $\mathcal{D}$ has a nullspace equal to span$\{1\}$, the constants. When solving exterior Dirichlet integral equations, then, we need to augment our representation in order to "patch up" this deficiency in $\mathcal{D}$. Normally, the modified layer potential is

$$\tilde{\mathcal{D}}\gamma = \mathcal{D}\gamma + \int_{\partial\Omega} \frac{1}{|x|}^{d-2} \gamma(x_0)\, dx_0, \tag{4.49}$$

where d is the dimension. For $d = 2$ this is $\mathcal{D}\gamma + c$, where c is a constant. However, this does not apply to our coupled FE-IE system, as we show next. Instead, we choose $\mathcal{A} = \mathcal{D} + \mathcal{S}$, the combination of the double and single layer with the same density, which has the added benefit of working in both two and three dimensions. With this representation, (4.47) becomes

$$\left\{ \frac{1}{2} I + (\mathcal{D}) + (\mathcal{S}) - \mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}} \right\} \gamma = g - \mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{M}}(v; \hat{g})\, f. \tag{4.50}$$

To understand the coupled FE-IE operator on the density γ, we first consider the following theorem from Kress [1]:

Theorem 4.1 (Kress, Theorem 2.16)**.** *Let X, Y, and Z be normed spaces and let $A : X \to Y$ and $B : Y \to Z$ be bounded linear operators. Then the product $BA : X \to Z$ is compact if one of the two operators A or B is compact.*

This states that the composition of a compact operator with bounded operators is still compact, which leads to our next theorem about the coupling operator $\mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}$.

Theorem 4.2. *Assume that $\hat{\Omega}$ is a convex domain or has a C^2 boundary and contains a smooth domain Ω. Then the operator $\mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}} : C(\partial\Omega) \to C(\partial\Omega)$, as defined previously, is compact.*

Proof. Consider the operator $\hat{\mathcal{R}} : C(\partial\Omega) \to C(\partial\hat{\Omega})$. Recall that $\hat{\mathcal{R}}\gamma$ evaluates the layer potential representation on the outer boundary $\partial\hat{\Omega}$, and that the layer potential kernels are continuous everywhere away from $\partial\Omega$. Therefore $\hat{\mathcal{R}}$ is an integral operator with continuous kernel and is therefore compact.

Next, consider $\hat{\mathcal{F}}^{-1}$ acting on $\hat{\mathcal{R}}\gamma$. As $\hat{\mathcal{R}}$ returns zero everywhere not on the boundary $\partial\hat{\Omega}$, $\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\gamma$ solves the weak harmonic problem with boundary conditions $\hat{\mathcal{R}}\gamma$ and is bounded due to the discrete weak maximum principle [62]. Due to the assumption on the domain $\hat{\Omega}$, we have elliptic regularity, and we can say that $\hat{\mathcal{F}}^{-1} : H^{1/2}(\partial\hat{\Omega}) \to H^2(\hat{\Omega})$. Thus, $\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\gamma \in H^2(\hat{\Omega})$ and is therefore continuous. Consequently, the pointwise restriction $\mathcal{R}$ exists and is bounded from $C(\hat{\Omega}) \to C(\partial\Omega)$. Together, we have $\mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}} : C(\partial\Omega) \to C(\partial\Omega)$ with $\mathcal{R}$ and $\hat{\mathcal{F}}^{-1}$ bounded and $\hat{\mathcal{R}}$ compact. By Theorem 4.1, it is compact. □

The result of Theorem 4.2 is that the total operator on γ, $\left\{\frac{1}{2}I + (\mathcal{D}) + (\mathcal{S}) - \mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\right\}$, is still of the form identity plus compact, and is still second kind. Thus by the Fredholm alternative, if the solution exists, it is also unique.

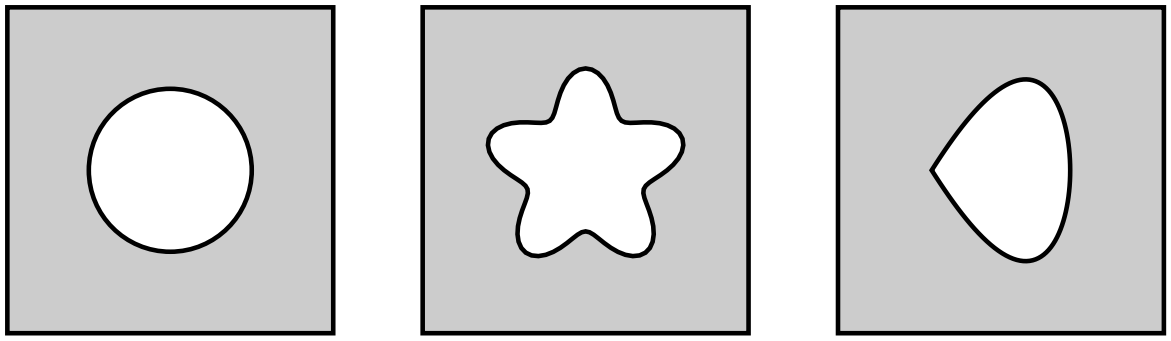
Examining the form of $\left\{\frac{1}{2}I + (\mathcal{D}) + (\mathcal{S}) - \mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\right\}$ highlights why the typical modified double layer potential (4.49) does not work in this case. The IE part of the operator is solving for a harmonic function outside Ω; the coupled FE part $\mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}$ is approximating a harmonic function interior to $\hat{\Omega}$ that has the same value as the u_2 on the boundary $\partial\hat{\Omega}$. If anything were in the range of both of these operators, they would cancel and create a nullspace in the equation (4.47). In the case of $\tilde{\mathcal{D}}$, this is exactly the case: a constant function given to $\hat{\mathcal{F}}^{-1}$ on the boundary $\partial\hat{\Omega}$ will give that same constant function everywhere in $\hat{\Omega}$, including $\partial\Omega$. This is demonstrated in Figures 4.5a and 4.5b, where we see the eigenvalues of a discrete approximation to three representations on three domains: a circle of radius $r = 0.5$, a *starfish* domain given by

$$\begin{aligned} \text{[starfish]} \qquad x(t) &= \frac{1}{2} + \frac{1}{8}\sin(10\pi t)\cos(2\pi t), \\ y(t) &= \frac{1}{2} + \frac{1}{8}\sin(10\pi t)\sin(2\pi t), \quad t \in [0,1], \end{aligned} \tag{4.51}$$

and a *drop* domain

$$\begin{aligned} \text{[drop]} \qquad x(t) &= \frac{17}{20}\left(\sin(\pi t) - \frac{1}{2}\right), \\ y(t) &= \frac{17}{20}\left(\cos(\pi t)\pi^2 t(t-1)\right), \quad t \in [0,1]. \end{aligned} \tag{4.52}$$

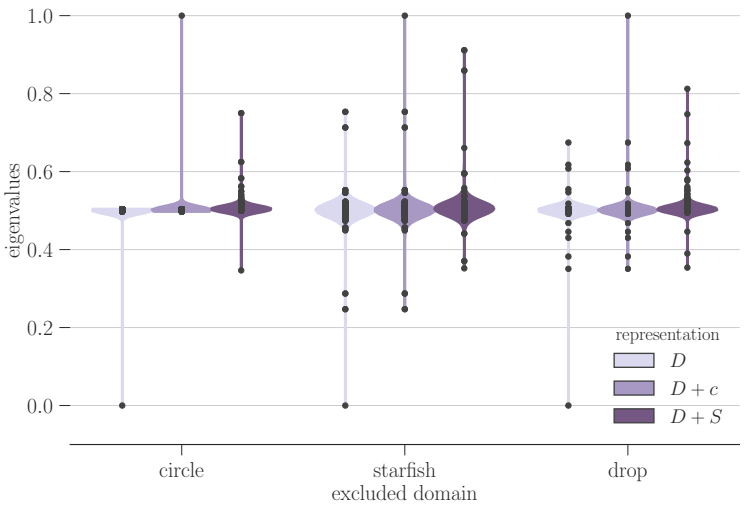
The three domains are shown in Figure 4.4. Although both $\mathcal{D} + c$ and $\mathcal{D} + \mathcal{S}$ remove the zero eigenvalue



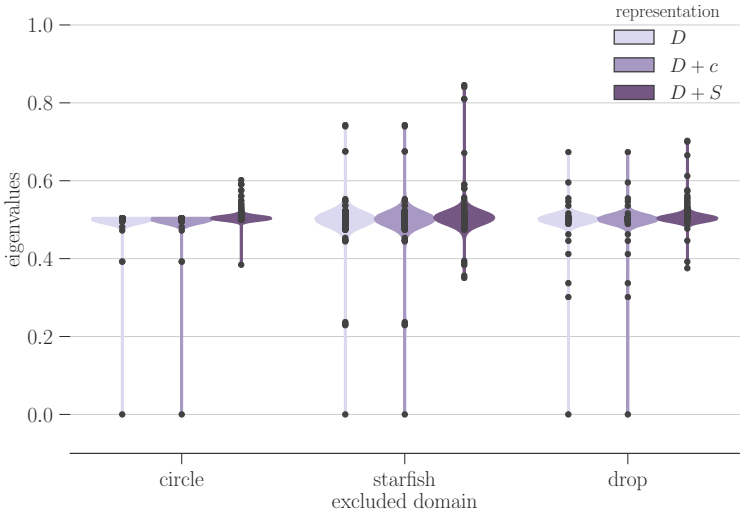
(a) Circle (b) Starfish (c) Drop

Figure 4.4: Three exterior domain examples.

from the $\mathcal{D}$ only representation, when we add the FE portion of the operator, the $\mathcal{D} + c$ case falls back to its zero eigenvalue.



(a) Eigenvalues for the IE-only operator.



(b) Eigenvalues for combined FE-IE operator.

Figure 4.5: Effects of a constant and single layer addition to the exterior double layer on three domains. The thickness of the colored areas indicates the width of the distributions of the eigenvalues.

Furthermore, if we examine the eigenvectors associated with the minimum eigenvalue for each representation in the coupled FE-IE operator, as shown in Figure 4.6, we see that the eigenvector for $\mathcal{D}$ and $\mathcal{D} + c$

is a constant for all three domains.

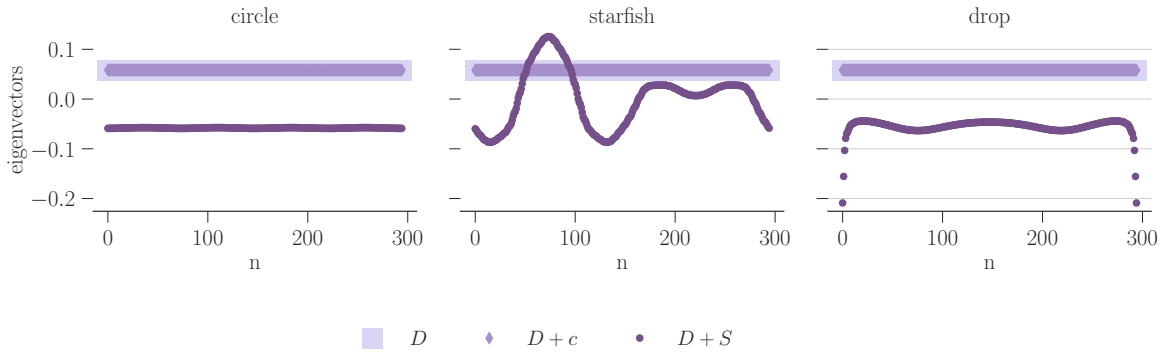


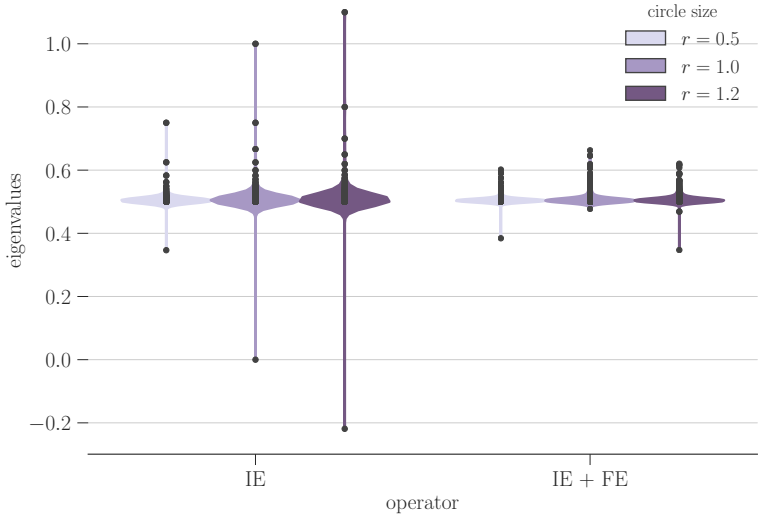
Figure 4.6: Eigenvectors associated with the minimum eigenvalue of the FE-IE operator for three domains.

There is an interesting phenomenon of the single layer in two dimensions that adds another facet to the $\mathcal{D}+\mathcal{S}$ operator. In three dimensions, the kernel of $\mathcal{S}$ does not change sign. However, in two dimensions, the Green's function is logarithmic and changes from positive to negative at $r = 1$. For example, $\mathcal{S}1$ is the sum of pairwise interactions among point charges with the same strength on the boundary $\partial\Omega$, and we see that some of the interactions are positive while others are negative when the domain is large enough; as a result, there is potential cancellation. Therefore, there could be certain domains for which a constant is in the nullspace of both $(\mathcal{D})$ and $(\mathcal{S})$, the on-boundary values. This is shown in Figure 4.7, where we see the minimum eigenvalue drop through zero to a negative value when increasing the domain size. However, in both cases, all eigenvalues of the combined FE-IE operator $\left\{\frac{1}{2}I + (\mathcal{D}) + (\mathcal{S}) - \mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}\right\}$ remain positive. This makes sense, as though $\mathcal{D}1$ is zero everywhere outside Ω, $\mathcal{S}1$ will grow toward infinity and will not be zero on the boundary $\partial\hat{\Omega}$.

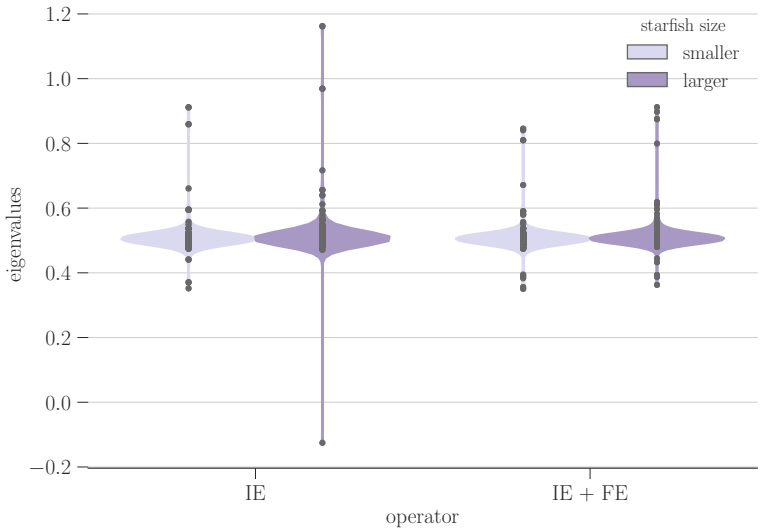
Error convergence

The error convergence is similar to the interior case, but with the additional composition of FE error and IE representation error in the operator $\mathcal{R}\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}}$. As it is a composition, however, we do not expect this error to change the overall order of convergence: the error in the $\hat{\mathcal{F}}^{-1}$ part of the overall operator on γ is bounded by the error in its boundary conditions — i.e., the operator error on $\hat{\mathcal{R}}$ — by the weak discrete maximum principle. Thus, the leading term effect of the composition of the two should be the same as the other FE or IE operator error terms, depending on which error is larger.

The finite element solution u_1 once again follows standard finite element convergence theory, with addi-



(a) Eigenvalues for the IE-only and combined FE-IE operator on three circle domains, with $\mathcal{A} = \mathcal{D} + \mathcal{S}$.



(b) Eigenvalues for the IE-only and combined FE-IE operator on two starfish domains, with $\mathcal{A} = \mathcal{D} + \mathcal{S}$.

Figure 4.7: Effects of a the size of Ω^i on the minimum eigenvalue of discrete approximations to $(\mathcal{D}) + (\mathcal{S})$.

tional error incurred through error in $\hat{\mathcal{R}}\gamma$ in the boundary condition of (4.48). However, again we know the extra FE error is bounded by the error in $\hat{\mathcal{R}}\gamma$ through the discrete weak maximum principle. The net result is that we expect to retain the same overall order of convergence as in the interior case, dependent on the FE and IE solvers in the same way.

Practical computation and discrete system

Now we discuss some additional details of the computation of the operators in (4.46) in a discrete setting. First, we introduce (D) and (S), the full matrices associated with the discretization of γ and computed through the approximation of $(\mathcal{D})$ and $(\mathcal{S})$. The discrete operator for $\mathcal{R}$ is an $m \times N$ matrix where m is the number of discretization points for the density γ and N is the number of degrees of freedom in the FE solution. Each row m corresponds to a point on the boundary $\partial\Omega$, and the columns are the values of the evaluated FE basis functions that are non-zero at that location. We denote this matrix R. Thus, if U_1 are the values at each degree of freedom for the FE solution u_1, then $RU_1 = r_1$.

The discrete operator for $\hat{\mathcal{R}}$ is an $N \times m$ matrix whose only rows with non-zero entries are the ones associated with degrees of freedom on the FE boundary $\partial\hat{\Omega}$. We denote this matrix as $\hat{R}$. Further define δ_j as a vector of length m which is equal to one for the jth element and zero everywhere else. Each non-zero row of $\hat{R}$ has m non-zero values with $\hat{R}_{ij}$ equal to the evaluation of $\mathcal{A}^h\delta_j$ at the location of FE boundary node i. Thus $\hat{R}\gamma^h = \hat{r}_2$.

Essential boundary conditions for the finite element problem

The coupling in (4.41) necessitates a deviation from the standard approach to handling essential boundary conditions in a finite element solver. When the boundary condition is known, the effects of the boundary are computed *a priori* and moved to the right hand side of the problem. In terms of the resulting matrix problem, this process is

1. Begin with the full matrix (associated with natural boundary conditions),
2. Remove rows associated with boundary degrees of freedom,
3. Apply the resulting matrix to a vector of the boundary condition evaluated at the boundary nodes and zero everywhere else,
4. Move the result of step 3 to the right hand side, and
5. Remove the columns associated with the boundary nodes from the matrix created in step 2.

For Galerkin finite elements, the final interior-only matrix is once again symmetric and positive definite.

However, in the case of the FE-IE coupling, the exact boundary condition for the FE solver is unknown, so the previous approach won't work. Instead, let F be the finite element matrix associated with the operator $\mathcal{F}(v)$ (4.14) for our chosen discrete basis, formulated with no imposed essential boundary conditions. Then define F_{int} as this same matrix but with the rows for boundary degrees of freedom set to zero. Let $\tilde{f}_{\text{int}}$ be

the modified right-hand side (as dictated by the weak form of the finite element problem) with boundary node entries zeroed. Finally, let $\hat{I}$ be the matrix whose entries are defined by $\hat{I}_{ij} = 1$ if $i = j$ and the ith degree of freedom is on $\partial\hat{\Omega}$, and $= 0$ otherwise. Thus, we see that our FE system with boundary conditions enforced is

$$\left\{F_{\text{int}} + \hat{I}\right\} U_1 = \tilde{f}_{\text{int}} + \hat{I}(\hat{g} - \hat{r}_2). \tag{4.53}$$

However, $\hat{r}_2$ depends on our unknown density γ. Thus, we can move this term to the left side and write

$$\hat{R}\gamma^h + \left\{F_{\text{int}} + \hat{I}\right\} U_1 = \tilde{f}_{\text{int}} + \hat{I}\hat{g}. \tag{4.54}$$

Finally we collect into our coupled discrete system:

$$\left[\begin{array}{c|c} \frac{1}{2}I + (D) + (S) & R \\ \hline \hat{R} & F_{\text{int}} + \hat{I} \end{array}\right] \left[\begin{array}{c} \gamma^h \\ \hline U_1 \end{array}\right] = \left[\begin{array}{c} g \\ \hline \tilde{f}_{\text{int}} + \hat{I}\hat{g} \end{array}\right]. \tag{4.55}$$

(D) and (S) are full matrices but only of dimension $m \times m$. All other matrices are sparse.

Demonstration of convergence

The numerical solution of the coupled system (4.55) is tested with the exact solution

$$u = \log(r_0) + 2\sin(\pi x)\sin(\pi y),$$

where

$$r_0 = \sqrt{(x - 0.1)^2 + (y + 0.02)^2}.$$

The same test function is solved on the circle, starfish (4.51), and drop (4.52) domains. The IE, FE, and total solutions are shown for the circular exclusion in Figure 4.8. Convergence results are found in Tables 4.2 and 4.3 for the circle and starfish domains, respectively. Once again, we observe high-order convergence.

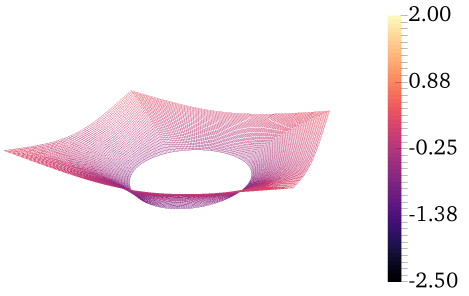
The convergence results for the drop domain are found in Table 4.4. Here the order of convergence is limited to ≈ 2 for the L^2 norm and ≈ 1 for the max norm. This is due to the corner in the drop domain, where the density has a singularity. The magnitude of error is for two cases meshes in Figure 4.9. While the error diminishes on the right side of the drop for the finer mesh, we see that error concentrated around the corner is still polluting the solution for a fairly large region, as expected from an elliptic problem.

p	qbx	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\hat{\Omega}\setminus\Omega)}$	order	$\|\text{error}\|_{L^2(\hat{\Omega}\setminus\Omega)}$	order
1	2	0.133, 0.224	5.11×10^{-2}	–	2.80×10^{-2}	–
		0.067, 0.108	1.15×10^{-2}	2.0	5.87×10^{-3}	2.1
		0.033, 0.053	2.90×10^{-3}	1.9	1.65×10^{-3}	1.8
		0.017, 0.026	7.25×10^{-4}	2.0	3.96×10^{-4}	2.0
2	3	0.133, 0.224	1.62×10^{-3}	–	1.15×10^{-3}	–
		0.067, 0.108	1.83×10^{-4}	3.0	1.39×10^{-4}	2.9
		0.033, 0.053	2.15×10^{-5}	3.0	1.66×10^{-5}	3.0
		0.017, 0.026	2.55×10^{-6}	3.0	2.20×10^{-6}	2.9
3	4	0.133, 0.224	2.02×10^{-4}	–	5.47×10^{-5}	–
		0.067, 0.108	9.55×10^{-6}	4.2	2.47×10^{-6}	4.3
		0.033, 0.053	4.84×10^{-7}	4.2	1.17×10^{-7}	4.3
		0.017, 0.026	2.35×10^{-8}	4.3	6.58×10^{-9}	4.1

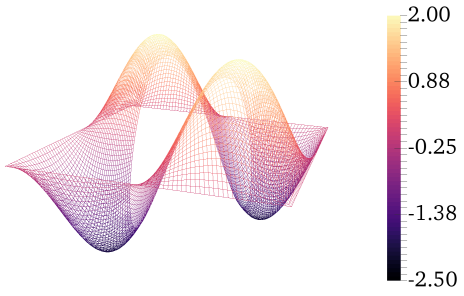
Table 4.2: Convergence of coupled interior-exterior FE-IE system for excluded circular domain.

p	qbx	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\hat{\Omega}\setminus\Omega)}$	order	$\|\text{error}\|_{L^2(\hat{\Omega}\setminus\Omega)}$	order
1	2	0.133, 0.327	5.80×10^{-2}	–	2.38×10^{-2}	–
		0.067, 0.170	1.28×10^{-2}	2.2	6.08×10^{-3}	2.0
		0.033, 0.085	3.47×10^{-3}	1.9	1.47×10^{-3}	2.0
		0.017, 0.043	8.46×10^{-4}	2.0	3.74×10^{-4}	2.0
2	3	0.133, 0.327	1.21×10^{-2}	–	2.05×10^{-3}	–
		0.067, 0.170	2.82×10^{-3}	2.1	3.29×10^{-4}	2.6
		0.033, 0.085	5.53×10^{-4}	2.3	3.70×10^{-5}	3.2
		0.017, 0.043	6.48×10^{-5}	3.1	3.65×10^{-6}	3.3
3	4	0.133, 0.327	1.00×10^{-2}	–	1.12×10^{-3}	–
		0.067, 0.170	1.73×10^{-3}	2.5	9.97×10^{-5}	3.5
		0.033, 0.085	1.91×10^{-4}	3.2	7.19×10^{-6}	3.8
		0.017, 0.043	1.47×10^{-5}	3.7	4.08×10^{-7}	4.1

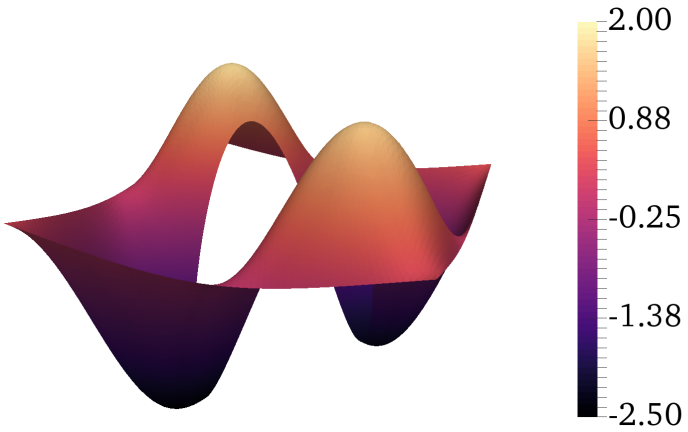
Table 4.3: Convergence of coupled interior-exterior FE-IE system for excluded starfish domain (4.51).



(a) IE solution on "true" domain.

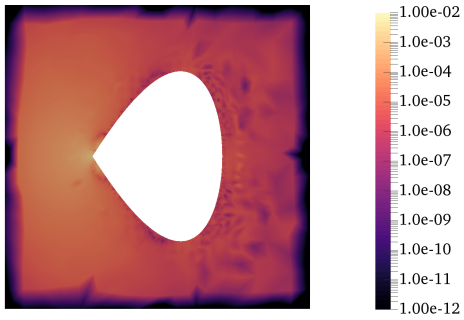


(b) FE solution on "true" domain.

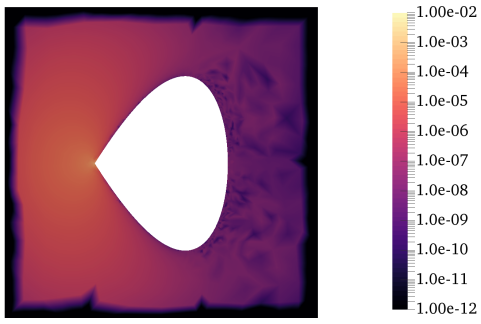


(c) Total solution.

Figure 4.8: Individual and combined solutions for the exterior embedded mesh problem.



(a) Error with h_{fe}, $h_{ie} = 0.067, 0.167$.



(b) Error with h_{fe}, $h_{ie} = 0.017, 0.042$.

Figure 4.9: Effect of domain corner on error. Solution shown for $p = 3$ and `qbx`$= 4$.

Numerical solution considerations

We consider two methods of solving the numerical system (4.55): solving the total system at once, and implementing the split solve technique introduced for the continuous problem for the purpose of analyzing the decomposition. In this case, we have the discrete version of (4.47 –4.48):

p	qbx	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\hat{\Omega}\setminus\Omega)}$	order	$\|\text{error}\|_{L^2(\hat{\Omega}\setminus\Omega)}$	order
1	2	0.133, 0.331	5.62×10^{-2}	–	2.76×10^{-2}	–
		0.067, 0.167	1.37×10^{-2}	2.0	6.89×10^{-3}	2.0
		0.033, 0.083	3.44×10^{-3}	2.0	1.71×10^{-3}	2.0
		0.017, 0.042	9.33×10^{-4}	1.9	4.29×10^{-4}	2.0
2	3	0.133, 0.331	4.95×10^{-3}	–	1.19×10^{-3}	–
		0.067, 0.167	2.29×10^{-3}	1.1	2.11×10^{-4}	2.5
		0.033, 0.083	1.10×10^{-3}	1.1	4.74×10^{-5}	2.2
		0.017, 0.042	5.12×10^{-4}	1.1	1.30×10^{-5}	1.9
3	4	0.133, 0.331	3.68×10^{-3}	–	3.48×10^{-4}	–
		0.067, 0.167	1.72×10^{-3}	1.1	9.79×10^{-5}	1.8
		0.033, 0.083	8.11×10^{-4}	1.1	2.84×10^{-5}	1.8
		0.017, 0.042	3.68×10^{-4}	1.1	8.29×10^{-6}	1.8

Table 4.4: Convergence of coupled interior-exterior FE-IE system for excluded drop domain (4.52).

1. Solve
$$\left\{\frac{1}{2}I + (D) + (S) - R[F_{\text{int}} + \hat{I}]^{-1}\hat{R}\right\}\gamma^h = g - R[F_{\text{int}} + \hat{I}]^{-1}(\tilde{f}_{\text{int}} + \hat{I}\hat{g}), \tag{4.56}$$
then

2. Solve
$$U_1 = [F_{\text{int}} + \hat{I}]^{-1}(\tilde{f}_{\text{int}} + \hat{I}\hat{g}) - [F_{\text{int}} + \hat{I}]^{-1}\hat{R}\gamma^h. \tag{4.57}$$

For the split solve, the advantage is that we regain the symmetric positive definiteness of the FE matrix that was lost by the non-conventional enforcement of Dirichlet boundary conditions. To do so, the degrees of freedom are reordered so that all the Dirichlet nodes are at the end. Then, since we do not want to actually compute $[F_{\text{int}} + \hat{I}]^{-1}$ explicitly, we will solve an iterative system when we need the action of the inverse. We note that after reordering, the matrix problem in (4.57) is

$$\left[\begin{array}{c|c} F_{\text{int-int}} & F_{\text{int-ext}} \\ \hline 0 & I \end{array}\right]\left[\begin{array}{c} U_{\text{int}} \\ \hline U_{\text{ext}} \end{array}\right] = \left[\begin{array}{c} \tilde{f} \\ \hline \hat{g} - \hat{R}\gamma^h \end{array}\right], \tag{4.58}$$

where $F_{\text{int-int}}$ is the finite element matrix coupling only interior nodes to other interior nodes — the matrix that results when we have known Dirichlet conditions and can remove the boundary nodes from the numerical system. Thus this is solved by setting $U_{\text{ext}} = \hat{g} - \hat{R}\gamma^h$ and iterating on

$$F_{\text{int-int}}U_{\text{int}} = \tilde{f} - F_{\text{int-ext}}(\hat{g} - \hat{R}\gamma^h),$$

which returns a symmetric positive definite matrix and allows the use of a preconditioned conjugate gradient (CG) method. We will call these interior FE problems the *inner* iterations for the split solve as they occur every time we need to apply the operator on γ^h to solve (4.56).

Preconditioning

To speed convergence of the numerical system (4.55), or the split solve (4.56–4.57), we apply preconditioning during our iterative solve. For the total system, we introduce the preconditioner

$$M_{\text{coup}} = \begin{bmatrix} I & 0 \\ 0 & M_{F_{\text{int}}+\hat{I}} \end{bmatrix} \tag{4.59}$$

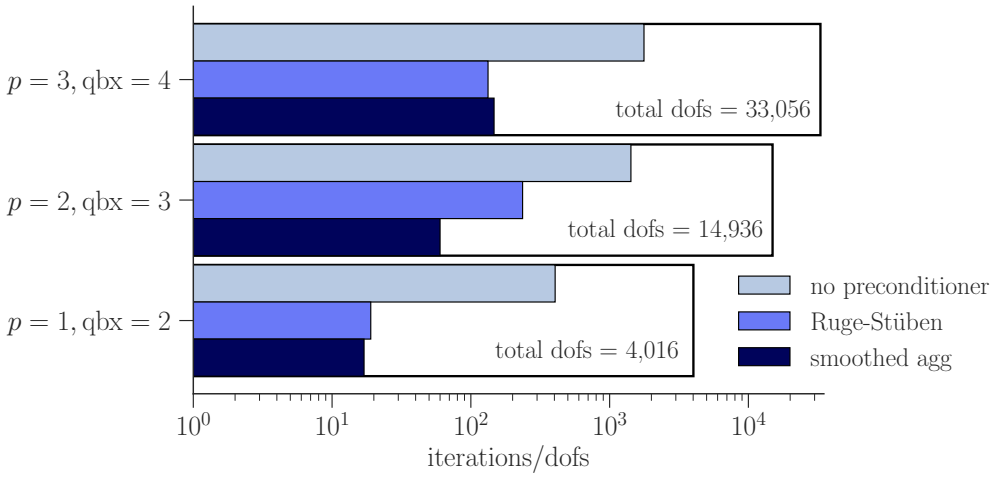
where $M_{F_{\text{int}}+\hat{I}}$ is the preconditioner for the matrix $F_{\text{int}} + \hat{I}$ alone. For the split solve, we precondition the inner iterations using $M_{F_{\text{int-int}}}$, the preconditioner for the interior-interior matrix.

In the following, we have used algebraic multigrid (AMG) as implemented in `pyamg` [63] to create preconditioners for these matrices. The effects of two types of AMG, Ruge-Stüben and smoothed aggregation, are shown for both the total and split solves. For the split solve, the average number of iterations per call to the inverse operator is reported. For both types of AMG, the maximum size of the coarse problem was set to 10.

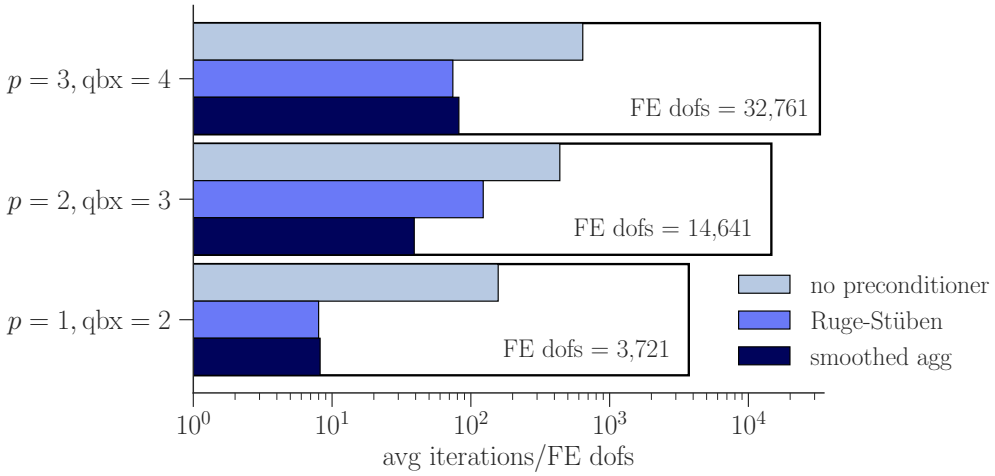
The effect of the preconditioners are shown in Figure 4.10 for the test problem with the circular domain Ω and h_{fe}, $h_{ie} = 0.033, 0.085$. The solve tolerance was $1{\times}10^{-12}$. We see that both types clearly improve convergence; in the rest of the numerical tests, we choose the smoothed aggregation preconditioner due to its better performance for $p = 2$ and better-or-comparable performance for $p = 1$ and $p = 3$.

Comparison of solution techniques

Finally, we also compare the total and split solve methods for the circular excluded domain. Figure 4.11a shows the required GMRES(20) iterations for the total system (4.55). Figure 4.11b shows both the outer GMRES iterations and the total inner CG iterations required across all calls to $F_{\text{int-int}}^{-1}$. We note that the number of outer iterations remains nearly constant across the different operator sizes and orders of basis functions. However, we see that the total number of CG iterations required is higher than the GMRES iterations for the associated total solve. For this problem and these discretizations, it is not clear that there is a significant advantage to using the split solve technique. This may change as the problem grows (in terms of number of degrees of freedom), especially as CG requires less memory than GMRES.

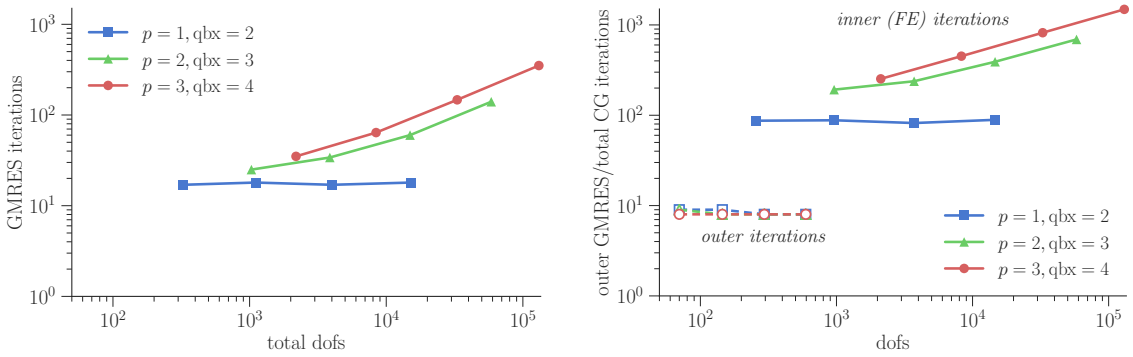


(a) Solving total coupled system with GMRES(20) and preconditioner M_{coup} (4.59).



(b) Effect of M_{FE} on the average number of iterations per call on $F_{\text{int-int}}^{-1}$.

Figure 4.10: Effects of two types of multigrid preconditioners in each solution type.



(a) Total solve

(b) Split solve

Figure 4.11: Comparison of solution techniques for the coupled system (4.46).

Chapter 5

FE-IE for interface problems

In this chapter, we combine the elements described in sections 4.2 and 4.3 to create a new method for embedded interface problems. The method requires no modification of basis functions at the interface and can handle a broad class of interface conditions, including both homogeneous and non-homogeneous jump conditions.

To develop the approach we consider the model problem

$$\begin{aligned} -\beta\nabla\cdot\nabla u(x) &= f(x) \quad \text{in} \quad \Omega^i \cup \Omega^e, \\ u^i(x) &= cu^e(x) + a(x) \quad \text{on} \quad \Gamma, \\ \frac{\partial u^i(x)}{\partial n} &= \kappa \frac{\partial u^e(x)}{\partial n} + b(x) \quad \text{on} \quad \Gamma \end{aligned} \tag{5.1}$$

where two domains Ω^i and Ω^e area separated by an interface Γ and the restriction of u to domain Ω^α written as u^α, with $\alpha = i, e$. This is illustrated in Figure 5.1. The coefficient β of the linear differential operator and the forcing function f may both be discontinuous across Γ.

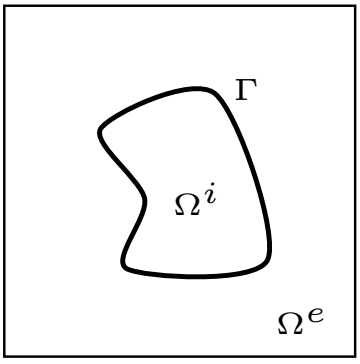


Figure 5.1: Two domains, Ω^i and Ω^e, separated by interface Γ.

This problem could describe, for example, steady-state heat conduction in two different objects or an object with changing material properties. In this new FE-IE approach, we introduce a fictitious domain $\hat{\Omega}^i$ such that $\Omega^i \subset \hat{\Omega}^i$, define $\hat{\Omega}^e$ as $\Omega^e \cup \Omega^i$, and split the problem into two subproblems with an appropriate

FE-IE splitting on each. Thus we have four components to our total solution: $u_1^i \in \hat{\Omega}^i$ and $u_2^i \in \Omega^i$ for the interior solution, plus $u_1^e \in \hat{\Omega}^e$ and $u_2^e \in \mathbb{R}^d \backslash \Omega^i$ for the exterior solution. This is illustrated in Figure 5.2.

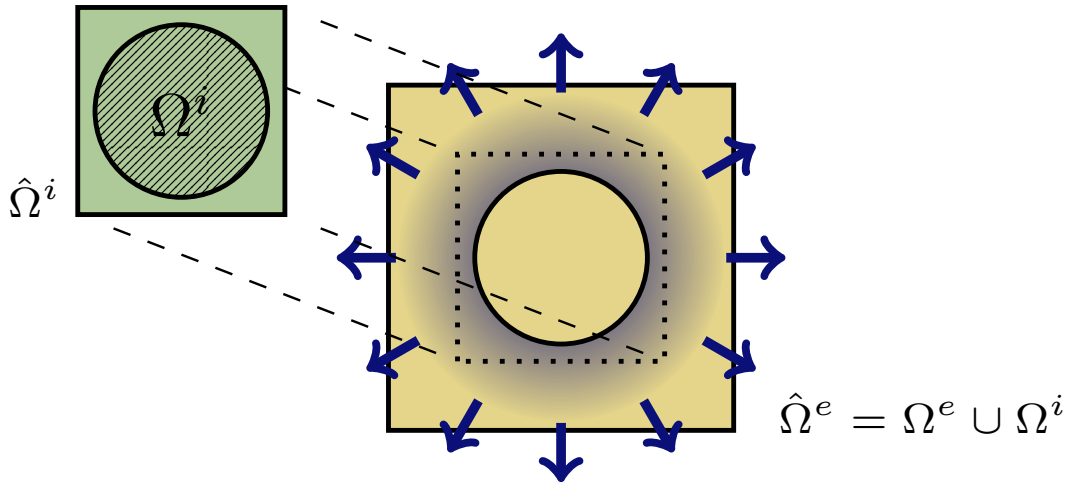


Figure 5.2: Coupled subproblems for an embedded interface.

5.1 Choice of layer potential representation

In this section, we discuss choices for the layer potential representations for u_2^i and u_2^e, and we show how to create second-kind integral equations for all but one possible combination of parameters κ and c.

Starting point: one layer potential each for u_2^i and u_2^e

The operators

$$-\frac{1}{2}I + (\mathcal{D}),$$

which arises from the interior Dirichlet problem with a double layer potential, and

$$-\frac{1}{2}I + (\mathcal{S}'),$$

which follows from the exterior Neumann problem, have trivial nullspaces and thus by Fredholm theory are guaranteed unique solutions to the associated integral equations. This motivates splitting (5.1) as

$$\begin{aligned}
\text{[Dirichlet FE]} \quad -\beta^i \nabla \cdot \nabla u_1^i &= f^i && \text{in} \quad \hat{\Omega}^i \\
u_1^i &= 0 && \text{on} \quad \partial\hat{\Omega}^i \qquad (5.2) \\
\text{[int. Dirichlet IE]} \quad -\nabla \cdot \nabla u_2^i &= 0 && \text{in} \quad \Omega^i \\
u_2^i &= c u_1^e + c u_2^e - u_1^i + a(x) && \text{on} \quad \Gamma \qquad (5.3) \\
\text{[coupled Dirichlet FE]} \quad -\beta^e \nabla \cdot \nabla u_1^e &= f^e && \text{in} \quad \hat{\Omega}^e \\
u_1^e &= \hat{g} - u_2^e && \text{on} \quad \partial\hat{\Omega}^e \qquad (5.4) \\
\text{[ext. Neumann IE]} \quad -\nabla \cdot \nabla u_2^e &= 0 && \text{in} \quad \mathbb{R}^2 \backslash \Omega^i \\
\kappa \frac{\partial u_2^e}{\partial n} &= \frac{\partial u_1^i}{\partial n} + \frac{\partial u_2^i}{\partial n} - \kappa \frac{\partial u_1^e}{\partial n} - b(x) && \text{on} \quad \Gamma. \qquad (5.5)
\end{aligned}$$

Then if we choose

$$u_2^i = \mathcal{D}\gamma^i$$

and

$$u_2^e = \mathcal{S}\gamma^e,$$

the resulting integral equations for u_2^i and u_2^e are

$$\begin{aligned}
\left\{ -\frac{1}{2} I + (\mathcal{D}) \right\} \gamma^i - (\mathcal{S})\gamma^e + \mathcal{R} u_1^i - c\mathcal{R} u_1^e &= a(x) \\
\kappa \left\{ -\frac{1}{2} I + (\mathcal{S}') \right\} \gamma^e + (\mathcal{D}')\gamma^i + \partial\mathcal{R} u_1^e - \kappa \partial\mathcal{R} u_1^e &= b(x), \qquad (5.6)
\end{aligned}$$

where we have introduced the operator $\partial\mathcal{R}$ to represent the normal derivative of u_1 at the interface. ($\mathcal{R}$ is as defined in the previous chapter.)

The key difficulty with this choice is the $(\mathcal{D}')$ operator in the derivative interface condition. It is not a compact operator and yields an equation that is no longer second kind. Indeed, having discretizations of $(\mathcal{D}')$ in a numerical system results in poor conditioning. Luckily, we can avoid this problem for all cases where $\kappa \neq c$, as we show next.

Improvement: the combination approach

We begin by using a general combined representation for the interior and exterior integral equation solutions, with

$$u_2^i = \alpha_1 \mathcal{D}\gamma^i + \alpha_2 \mathcal{S}\gamma^e, \tag{5.7}$$

and

$$u_2^e = \alpha_3 \mathcal{D}\gamma^i + \alpha_4 \mathcal{S}\gamma^e, \tag{5.8}$$

for some constant coefficients α_j. This will allow us to pick the α_j such that all integral operators are well-conditioned.

Taking the limits of these expressions at the interface and adding the interface restrictions of the finite element solutions gives the following form for the interface conditions:

$$\mathcal{R}^i u_1^i + \alpha_1 \mathcal{D}_- \gamma^i + \alpha_2 \mathcal{S}_- \gamma^e = c\partial\mathcal{R}^e u_1^e + c\alpha_3 \mathcal{D}_+ \gamma^i + c\alpha_4 \mathcal{S}_+ \gamma^e + a(x), \tag{5.9}$$

and

$$\partial\mathcal{R}^i u_1^i + \alpha_1 \partial_n \mathcal{D}_- \gamma^i + \alpha_2 \partial_n \mathcal{S}_- \gamma^e = \kappa\partial\mathcal{R}^e u_1^e + \kappa\alpha_3 \partial_n \mathcal{D}_+ \gamma^i + \kappa\alpha_4 \partial_n \mathcal{S}_+ \gamma^e + b(x), \tag{5.10}$$

where $\mathcal{D}_\pm$ indicates the interior $(-)$ or exterior $(+)$ limit of the double-layer operator, and likewise for $\mathcal{S}_\pm$. Due to the known jump conditions of the single- and double-layer operators (4.6–4.9), we can rewrite these expressions as

$$\mathcal{R}^i u_1^i + \alpha_1 \left(-\frac{1}{2}\gamma^i + (\mathcal{D})\gamma^i \right) + \alpha_2 (\mathcal{S})\gamma^e = c\mathcal{R}^e u_1^e + c\alpha_3 \left(\frac{1}{2}\gamma^i + (\mathcal{D})\gamma^i \right) + c\alpha_4 (\mathcal{S})\gamma^e + a(x) \tag{5.11}$$

and

$$\partial\mathcal{R}^i u_1^i + \alpha_1 (\mathcal{D}')\gamma^i + \alpha_2 \left(\frac{1}{2}\gamma^e + (\mathcal{S}')\gamma^e \right) = \kappa\partial\mathcal{R}^e u_1^e + \kappa\alpha_3 (\mathcal{D}')\gamma^i + \kappa\alpha_4 \left(-\frac{1}{2}\gamma^e + (\mathcal{S}')\gamma^e \right) + b(x), \tag{5.12}$$

where we recall that $(\mathcal{D})$ and $(\mathcal{S}')$ now indicate the on-boundary value of the double-layer operator and normal derivative of the single-layer operator, respectively, which are defined as the Cauchy principal values.

Collecting the terms on each operator yields

$$\mathcal{R}^i u_1^i + \left[-\frac{\alpha_1 + c\alpha_3}{2} I + (\alpha_1 - c\alpha_3)(\mathcal{D}) \right] \gamma^i = c\mathcal{R}^e u_1^e + \left[-\alpha_2 + c\alpha_4 \right] (\mathcal{S})\gamma^e + a(x) \tag{5.13}$$

and

$$\partial\mathcal{R}^i u_1^i + \left[\frac{\alpha_2 + \kappa\alpha_4}{2} I + (\alpha_2 - \kappa\alpha_4)(\mathcal{S}')\right] \gamma^e = \kappa\partial\mathcal{R}^e u_1^e + \left[-\alpha_1 + \kappa\alpha_3\right](\mathcal{D}')\gamma^i + b(x). \tag{5.14}$$

This form of the conditions is used to determine the best values for the α_j coefficients. First, we wish to eliminate the hypersingular operator $(\mathcal{D}')$ from the derivative condition, our original goal in using this combined representation. Thus, we see that

$$\alpha_1 = \kappa\alpha_3 \tag{5.15}$$

is our first requirement. With $(\mathcal{D}')$ eliminated, (5.14) is an equation for γ^e with operator

$$\frac{\alpha_2 + \kappa\alpha_4}{2} I + (\alpha_2 - \kappa\alpha_4)(\mathcal{S}').$$

In order to have an operator with only the trivial nullspace, guaranteeing a unique solution for γ^e, we request that the coefficients of I and $(\mathcal{S}')$ have opposite sign.

We return to the value jump condition (5.13). Enforcing the requirement for α_1 and α_3 (5.15), the operator on γ^i is

$$-\frac{(\kappa + c)\alpha_3}{2} I + (\kappa - c)\alpha_3(\mathcal{D}). \tag{5.16}$$

Here we have three possibilities based on κ and c:

1. $\kappa \neq c$ and $\kappa \neq -c$,
2. $\kappa = c$, where the $(\mathcal{D})$ term is cancelled, and
3. $\kappa = -c$, where the identity term is cancelled.

We will now consider the three cases separately.

The case $\kappa \neq c$ and $\kappa \neq -c$

In this case, after applying (5.15) and adding equations for the FE solutions, the resulting coupled system of equations is described by:

Find $\gamma^i, \gamma^e, u_1^i, u_1^e$ *such that*

$$\left[\begin{array}{cc|cc} -\frac{1}{2}(\kappa+c)\alpha_3 I + (\kappa-c)\alpha_3(\mathcal{D}) & \mathcal{R} & (\alpha_2 - c\alpha_4)(\mathcal{S}) & -c\mathcal{R} \\ 0 & \mathcal{F}(v)^i & 0 & 0 \\ \hline 0 & \partial\mathcal{R} & \frac{1}{2}(\alpha_2+\kappa\alpha_4)I + (\alpha_2-\kappa\alpha_4)(\mathcal{S}') & -\kappa\partial\mathcal{R} \\ \alpha_3\hat{\mathcal{R}}_{\mathcal{D}} & 0 & \alpha_4\hat{\mathcal{R}}_{\mathcal{S}} & \hat{\mathcal{F}}(w)^e \end{array}\right] \left[\begin{array}{c} \gamma^i \\ u_1^i \\ \hline \gamma^e \\ u_1^e \end{array}\right] = \left[\begin{array}{c} a(x) \\ \mathcal{M}(v)\, f^i \\ \hline b(x) \\ \hat{\mathcal{M}}(w;\hat{g})\, f^e \end{array}\right] \tag{5.17}$$

$\forall v \in H_0^1(\hat{\Omega}^i)$ and $w \in H_0^1(\hat{\Omega}^e)$. The interior FE problem can have any boundary conditions we choose, and therefore does not need the special modifications discussed previously and denoted by the notation changes for $\mathcal{F}$ and $\mathcal{M}$. For notation, $\hat{\mathcal{R}}_{\mathcal{D}}$ signifies that it is the $\hat{\mathcal{R}}$ operator for the double layer only, and likewise for the single layer and $\hat{\mathcal{R}}_{\mathcal{S}}$.

An important point to note is that any added jump conditions $a(x)$ and $b(x)$ are simply added to the right hand side. Thus homogeneous and non-homogeneous jump conditions are *exactly the same*, and do not require any changes to the matrix.

Split solve analysis

As for the exterior problem, we also reorder the system (5.17) and consider its properties using a Schur complement approach. Instead of grouping by interior and exterior, we group by IE and FE:

$$\left[\begin{array}{cc|cc} -\frac{1}{2}(\kappa+c)\alpha_3 I + (\kappa-c)\alpha_3(\mathcal{D}) & (\alpha_2 - c\alpha_4)(\mathcal{S}) & \mathcal{R} & -c\mathcal{R} \\ 0 & \frac{1}{2}(\alpha_2+\kappa\alpha_4)I + (\alpha_2-\kappa\alpha_4)(\mathcal{S}') & \partial\mathcal{R} & -\kappa\partial\mathcal{R} \\ \hline 0 & 0 & \mathcal{F}(v)^i & 0 \\ \alpha_3\hat{\mathcal{R}}_{\mathcal{D}} & \alpha_4\hat{\mathcal{R}}_{\mathcal{S}} & 0 & \hat{\mathcal{F}}(w)^e \end{array}\right] \left[\begin{array}{c} \gamma^i \\ \gamma^e \\ \hline u_1^i \\ u_1^e \end{array}\right] = \left[\begin{array}{c} a(x) \\ b(x) \\ \hline \mathcal{M}(v)\, f^i \\ \hat{\mathcal{M}}(w;\hat{g})\, f^e \end{array}\right]. \tag{5.18}$$

To simplify notation, we define some names for the blocks:

$$\underline{\underline{\mathcal{A}}} = \begin{bmatrix} -\frac{1}{2}(\kappa + c)\alpha_3 I + (\kappa - c)\alpha_3(\mathcal{D}) & (\alpha_2 - c\alpha_4)(\mathcal{S}) \\ 0 & \frac{1}{2}(\alpha_2 + \kappa\alpha_4)I + (\alpha_2 - \kappa\alpha_4)(\mathcal{S}') \end{bmatrix}, \tag{5.19}$$

$$\underline{\underline{\mathcal{R}}} = \begin{bmatrix} \mathcal{R} & -c\mathcal{R} \\ \partial\mathcal{R} & -\kappa\partial\mathcal{R} \end{bmatrix}, \tag{5.20}$$

$$\underline{\underline{\mathcal{F}}} = \begin{bmatrix} \mathcal{F}(v)^i & 0 \\ 0 & \hat{\mathcal{F}}(w)^e \end{bmatrix}, \tag{5.21}$$

and

$$\underline{\underline{\hat{\mathcal{R}}}} = \begin{bmatrix} 0 & 0 \\ \alpha_3\hat{\mathcal{R}}_{\mathcal{D}} & \alpha_4\hat{\mathcal{R}}_{\mathcal{S}} \end{bmatrix}. \tag{5.22}$$

The resulting decomposition of the solution is written as

1. Solve

$$\left\{\underline{\underline{\mathcal{A}}} - \underline{\underline{\mathcal{R}}}\,\underline{\underline{\mathcal{F}}}^{-1}\underline{\underline{\hat{\mathcal{R}}}}\right\}\begin{bmatrix}\gamma^i \\ \gamma^e\end{bmatrix} = \begin{bmatrix} a(x) \\ b(x)\end{bmatrix} - \underline{\underline{\mathcal{R}}}\,\underline{\underline{\mathcal{F}}}^{-1}\begin{bmatrix}\mathcal{M}(v)\,f^i \\ \hat{\mathcal{M}}(v;\hat{g})\,f^e\end{bmatrix}, \tag{5.23}$$

   and then

2. Evaluate

$$\begin{bmatrix} u_1^i \\ u_1^e \end{bmatrix} = \underline{\underline{\mathcal{F}}}^{-1}\left(\begin{bmatrix}\mathcal{M}(v)\,f^i \\ \hat{\mathcal{M}}(w;\hat{g})\,f^e\end{bmatrix} - \underline{\underline{\hat{\mathcal{R}}}}\begin{bmatrix}\gamma^i \\ \gamma^e\end{bmatrix}\right). \tag{5.24}$$

Due to the nature of $\underline{\underline{\hat{\mathcal{R}}}}$, only the exterior FE problem affects the final operator on the densities γ^i and γ^e. That is,

$$\underline{\underline{\mathcal{F}}}^{-1}\underline{\underline{\hat{\mathcal{R}}}} = [\mathcal{F}^e]^{-1}(\alpha_3\hat{\mathcal{R}}_{\mathcal{D}}\gamma^i + \alpha_4\hat{\mathcal{R}}_{\mathcal{S}}\gamma^e).$$

The equations (5.23–5.24) very closely resemble the split solve equations for the exterior case in Section 4.3. The main difference here — apart from the doubling of the number of variables — is the appearance of the $\partial\mathcal{R}$ operators. As the output from $\underline{\underline{\hat{\mathcal{R}}}}\boldsymbol{\gamma}$ is smooth, however, we expect that for most domains $\Omega^e \cup \Omega^i$, the result of $[\mathcal{F}^e]^{-1}(\alpha_3\hat{\mathcal{R}}_{\mathcal{D}}\gamma^i + \alpha_4\hat{\mathcal{R}}_{\mathcal{S}}\gamma^e)$ is smooth, as it is approximating a harmonic function.

Numerical computation

The discrete system corresponding to (5.17) can be written

$$\begin{bmatrix} -\frac{1}{2}(\kappa+c)\alpha_3 I + (\kappa-c)\alpha_3 (D)^{i\to i} & R^{i\to i} & (\alpha_2 - c\alpha_4)(S)^{e\to i} & -cR^{e\to i} \\ 0 & \mathcal{F}^i & 0 & 0 \\ 0 & \partial R^{i\to e} & \frac{1}{2}(\alpha_2+\kappa\alpha_4)I + (\alpha_2-\kappa\alpha_4)(S')^{e\to e} & -\kappa\partial R^{e\to e} \\ \alpha_3 \hat{R}_D^i & 0 & \alpha_4 \hat{R}_S^e & \mathcal{F}_{\text{int}}^e + \hat{I} \end{bmatrix} \begin{bmatrix} \gamma^{i^h} \\ U_1^i \\ \gamma^{e^h} \\ U_1^e \end{bmatrix} = \begin{bmatrix} a(x) \\ \tilde{f}^i \\ b(x) \\ \tilde{f}^e + \hat{I}\hat{g} \end{bmatrix} \tag{5.25}$$

The main detail necessary for computation of the discrete operators is represented by superscript $i \to e$, which denotes an operation on the interior variable evaluated at the computational nodes for the exterior variable. The new matrix $\partial R^{i\to e}$ is defined in the same way as $R^{i\to e}$, but for the normal derivatives of the FE basis functions on the interface Γ.

The case $\kappa = c$

In this case, the first line of (5.17) is replaced by

$$\begin{bmatrix} -\frac{1}{2}(\kappa+c)\alpha_3 I & \mathcal{R} & (\alpha_2 - c\alpha_4)(\mathcal{S}) & -c\mathcal{R} \end{bmatrix}. \tag{5.26}$$

Furthermore, we can choose $\alpha_2 = c\alpha_4 = \kappa\alpha_4$, meaning the first and third lines of the system are

$$\begin{bmatrix} -\frac{1}{2}(\kappa+c)\alpha_3 I & \mathcal{R} & 0 & -c\mathcal{R} \end{bmatrix} \tag{5.27}$$

and

$$\begin{bmatrix} 0 & \partial\mathcal{R} & \kappa\alpha_2 I & -\kappa\partial\mathcal{R} \end{bmatrix}. \tag{5.28}$$

If we consider the effect on the split solve operators, we see immediately how advantageous this case is: the block $\underline{\underline{\mathcal{A}}}$ is now

$$\underline{\underline{\mathcal{A}}}_{\kappa=c} = \begin{bmatrix} -\frac{1}{2}(\kappa+c)\alpha_3 I & 0 \\ 0 & \kappa\alpha_2 I \end{bmatrix}. \tag{5.29}$$

The case $\kappa = -c$

In this case, the operator on γ^i as defined in (5.16) loses the identity term, resulting in an equation for γ^i which is not second kind. We leave this to future work.

5.2 Numerical tests

We investigate convergence results with three test problems, two with $\kappa \neq c$ and one with $\kappa = c$. The specific choices for the coefficients of the layer potential representation are summarized in Table 5.1.

	$\kappa \neq c$	$\kappa = c$
α_1	$\kappa\alpha_3$	$\kappa\alpha_3$
α_2	0	1
α_3	$1/(\kappa - c)$	$-1/\kappa$
α_4	$1/\kappa$	$1/\kappa$

Table 5.1: Choices of coefficients for numerical tests.

The "quadratic-log" test case

This case is a manufactured solution for the case when the interface Γ is a circle. Specifically, we consider Γ to be a circle of radius 0.5 and $\Omega^e \cup \Omega^i$ to be the square $[-1, -1] \times [1, 1]$. The definition of this test case is given in Table 5.2 and it is shown in Figure 5.3. The error convergence for two different sizes of $\hat{\Omega}^i$ can be

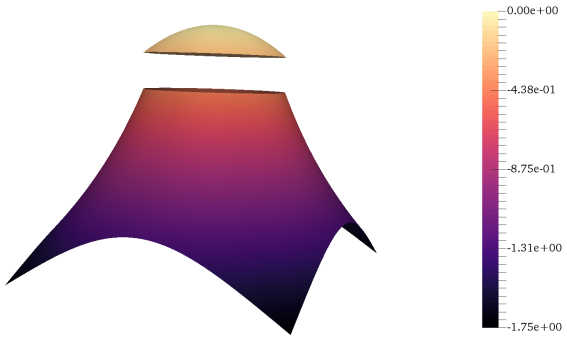


Figure 5.3: Total solution for the "quadratic-log" test case

seen in Tables 5.3 and 5.4. We note that in both cases, we achieve high-order convergence for basis functions $p = 2$ and above. While slightly lower than for the exterior problem or interior problem alone, we attribute this to the fact that the solution now depends on the representation of the FE derivative, and we expect to

	interior	exterior	interface conditions
exact solution	$-\frac{5}{6}(x^2+y^2)$	$-\frac{5}{4}\log\left(\frac{1}{2r}\right)-\frac{11}{24}$ $(r=\sqrt{x^2+y^2})$	$\kappa = 1/3$
coefficient, β	3	1	$c = 1$
r.h.s, f	10	0	$a(x) = 1/4$
boundary cond.	*N/A*	*exact soln*	$b(x) = 0$

Table 5.2: Description of "quadratic-log" case.

lose an order of convergence in the gradient representation compared to the solution representation. The extent to which this affects the overall solution changes, but should reach a limit at losing the entire order in the overall solution.

An interesting artifact arises in the convergence for $p = 1$: for the smaller $\hat{\Omega}^i$ domain, there is negative convergence between the coarsest and second-coarsest meshes. This is also due to the appearance of the derivative of the finite element solution in the system. For $p = 1$, the derivative of the FE solution is piecewise constant. On a coarse mesh, piecewise constants poorly represent even smooth solutions with variation. As a result, the $p = 1$ case is very sensitive to element placement, especially for coarse meshes. Because of this, as a general rule, it is recommended to use at least $p = 2$. We also observe that for $p = 2$ and higher, there is little difference between the results for the two domain sizes.

As another measure of convergence, we consider how well we are satisfying the specified conditions on the interface. The convergence of the error in the interface conditions for the two quadratic-log problems are shown in Tables 5.5 and 5.6. For all cases, the norms were approximated on the same fourth-order boundary mesh with 124 elements. For both cases of $\hat{\Omega}^i$, we see high-order convergence in the interface conditions. The value condition converges approximately at a rate of $p + 1$, and we lose an order of convergence in the derivative, as expected.

The "general jump" test case

Now we consider a second case for which $\kappa \neq c$. In this case, there is no exact solution. It is similar to the quadratic-log case, but with different values for β^i and κ, and with homogeneous Dirichlet boundary conditions enforced on $\partial\Omega^e$. The case is summarized in Table 5.7 and shown in Figure 5.4. The IE and FE components for a sample solution are shown in Figure 5.5. As there is no exact solution for this case, we cannot compute the error throughout the domains. However, we still compute the error in the interface conditions and check their convergence rates. This is summarized in Tables 5.8 and 5.9 for a circular and starfish interface, respectively.

p	qbx	solution	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\Omega)}$	order	$\|\text{error}\|_{L^2(\Omega)}$	order
1	2	interior	0.080, 0.224	6.93×10^{-3}	–	3.81×10^{-3}	–
			0.040, 0.108	1.43×10^{-2}	-1.0	1.14×10^{-2}	-1.5
			0.020, 0.053	5.48×10^{-3}	1.4	3.88×10^{-3}	1.5
			0.010, 0.026	1.51×10^{-3}	1.8	8.11×10^{-4}	2.2
1	2	exterior	0.080, 0.224	8.53×10^{-3}	–	5.61×10^{-3}	–
			0.040, 0.108	1.51×10^{-2}	-0.8	1.05×10^{-2}	-0.9
			0.020, 0.053	5.34×10^{-3}	1.5	3.43×10^{-3}	1.6
			0.010, 0.026	1.48×10^{-3}	1.8	7.23×10^{-4}	2.2
2	3	interior	0.080, 0.224	3.64×10^{-3}	–	2.57×10^{-3}	–
			0.040, 0.108	6.41×10^{-4}	2.4	3.88×10^{-4}	2.6
			0.020, 0.053	7.57×10^{-5}	3.0	4.34×10^{-5}	3.1
			0.010, 0.026	1.05×10^{-5}	2.8	5.64×10^{-6}	2.9
2	3	exterior	0.080, 0.224	5.14×10^{-3}	–	2.63×10^{-3}	–
			0.040, 0.108	7.79×10^{-4}	2.6	3.78×10^{-4}	2.7
			0.020, 0.053	8.63×10^{-5}	3.1	4.15×10^{-5}	3.1
			0.010, 0.026	1.12×10^{-5}	2.9	5.27×10^{-6}	2.9
3	4	interior	0.080, 0.224	7.95×10^{-4}	–	4.22×10^{-4}	–
			0.040, 0.108	8.04×10^{-5}	3.1	3.83×10^{-5}	3.3
			0.020, 0.053	6.34×10^{-6}	3.6	2.75×10^{-6}	3.7
			0.010, 0.026	4.52×10^{-7}	3.8	1.91×10^{-7}	3.8
3	4	exterior	0.080, 0.224	1.16×10^{-3}	–	4.77×10^{-4}	–
			0.040, 0.108	9.99×10^{-5}	3.4	3.99×10^{-5}	3.4
			0.020, 0.053	7.13×10^{-6}	3.7	2.77×10^{-6}	3.8
			0.010, 0.026	4.80×10^{-7}	3.8	1.87×10^{-7}	3.8

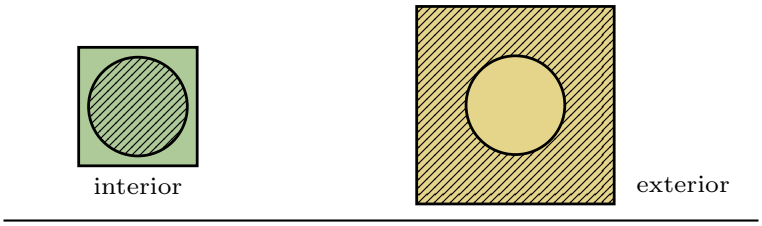
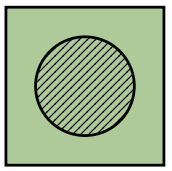


Table 5.3: Error convergence for quadratic-log test problem with $\hat{\Omega}^i = [-0.6, -0.6] \times [0.6, 0.6]$.

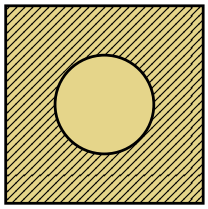
The "sine-linear" test case

Finally, we consider a test case for which $\kappa = c$; it is summarized in Table 5.10 and shown for a circular interface in Figure 5.6. This is also a manufactured solution, now with the same forcing function f and coefficient β through $\Omega^i \cup \Omega^e$. The extra linear function added to the interior solution influences the numerical system only through the non-homogeneous jump conditions $a(x)$ and $b(x)$. This case is tested on both the circular and starfish interfaces. The error convergence is shown in Tables 5.11 and 5.12, and the interface condition errors are shown in Tables 5.13 and 5.14. Again, we observe high-order convergence. However, there is an interesting phenomenon seen, for example, in the $p = 2$ rows of Table 5.12. The convergence

p	qbx	solution	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\Omega)}$	order	$\|\text{error}\|_{L^2(\Omega)}$	order
1	2	interior	0.080, 0.224	2.69×10^{-2}	–	1.90×10^{-2}	–
			0.040, 0.108	1.25×10^{-2}	1.1	9.99×10^{-3}	0.9
			0.020, 0.053	4.63×10^{-3}	1.4	3.27×10^{-3}	1.6
			0.010, 0.026	1.18×10^{-3}	1.9	5.94×10^{-4}	2.4
1	2	exterior	0.080, 0.224	3.06×10^{-2}	–	1.95×10^{-2}	–
			0.040, 0.108	1.30×10^{-2}	1.2	9.22×10^{-3}	1.0
			0.020, 0.053	4.53×10^{-3}	1.5	2.88×10^{-3}	1.6
			0.010, 0.026	1.16×10^{-3}	1.9	5.28×10^{-4}	2.4
2	3	interior	0.080, 0.224	2.79×10^{-3}	–	2.12×10^{-3}	–
			0.040, 0.108	5.48×10^{-4}	2.2	3.82×10^{-4}	2.4
			0.020, 0.053	6.81×10^{-5}	2.9	4.67×10^{-5}	3.0
			0.010, 0.026	9.30×10^{-6}	2.8	6.14×10^{-6}	2.9
2	3	exterior	0.080, 0.224	3.76×10^{-3}	–	2.18×10^{-3}	–
			0.040, 0.108	6.35×10^{-4}	2.4	3.65×10^{-4}	2.5
			0.020, 0.053	7.47×10^{-5}	3.0	4.34×10^{-5}	3.0
			0.010, 0.026	9.72×10^{-6}	2.9	5.60×10^{-6}	2.9
3	4	interior	0.080, 0.224	6.56×10^{-4}	–	4.21×10^{-4}	–
			0.040, 0.108	6.36×10^{-5}	3.2	3.71×10^{-5}	3.3
			0.020, 0.053	4.99×10^{-6}	3.6	2.70×10^{-6}	3.7
			0.010, 0.026	3.48×10^{-7}	3.8	1.82×10^{-7}	3.8
3	4	exterior	0.080, 0.224	8.94×10^{-4}	–	4.40×10^{-4}	–
			0.040, 0.108	7.61×10^{-5}	3.4	3.61×10^{-5}	3.4
			0.020, 0.053	5.50×10^{-6}	3.7	2.55×10^{-6}	3.7
			0.010, 0.026	3.66×10^{-7}	3.9	1.69×10^{-7}	3.9



interior



exterior

Table 5.4: Error convergence for quadratic-log test problem with $\hat{\Omega}^i = [-0.8, -0.8] \times [0.8, 0.8]$.

orders decline from 4 to 2 as the meshes are refined. This is likely the effect of the form of the error for the interior and exterior problems, where total error is bounded by a sum of IE and FE error terms. A change in which error term dominates would result in a change in the overall apparent convergence if the terms are not all the same order. As we have introduced the FE derivative to the IE boundary conditions and thus the IE error, we have an error term that is one order lower than the rest.

p	qbx	condition	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\partial\Omega)}$	order	$\|\text{error}\|_{L^2(\partial\Omega)}$	order
1	2	value	0.080, 0.224	8.69×10^{-3}	–	8.00×10^{-3}	–
			0.040, 0.108	1.40×10^{-3}	2.5	1.42×10^{-3}	2.4
			0.020, 0.053	2.82×10^{-4}	2.3	2.25×10^{-4}	2.6
			0.010, 0.026	6.21×10^{-5}	2.2	3.96×10^{-5}	2.5
		derivative	0.080, 0.224	9.78×10^{-2}	–	7.21×10^{-2}	–
			0.040, 0.108	6.66×10^{-2}	0.5	4.00×10^{-2}	0.8
			0.020, 0.053	3.19×10^{-2}	1.0	1.86×10^{-2}	1.1
			0.010, 0.026	1.60×10^{-2}	1.0	9.35×10^{-3}	1.0
2	3	value	0.080, 0.224	1.60×10^{-3}	–	1.86×10^{-3}	–
			0.040, 0.108	1.55×10^{-4}	3.2	1.77×10^{-4}	3.2
			0.020, 0.053	1.28×10^{-5}	3.5	1.40×10^{-5}	3.6
			0.010, 0.026	9.05×10^{-7}	3.8	9.90×10^{-7}	3.8
		derivative	0.080, 0.224	1.17×10^{-2}	–	1.21×10^{-2}	–
			0.040, 0.108	2.78×10^{-3}	2.0	2.84×10^{-3}	2.0
			0.020, 0.053	4.86×10^{-4}	2.5	4.70×10^{-4}	2.5
			0.010, 0.026	7.89×10^{-5}	2.6	6.86×10^{-5}	2.7
3	4	value	0.080, 0.224	4.21×10^{-4}	–	4.77×10^{-4}	–
			0.040, 0.108	2.32×10^{-5}	4.0	2.59×10^{-5}	4.0
			0.020, 0.053	9.95×10^{-7}	4.4	1.09×10^{-6}	4.5
			0.010, 0.026	3.63×10^{-8}	4.7	3.95×10^{-8}	4.7
		derivative	0.080, 0.224	4.96×10^{-3}	–	5.66×10^{-3}	–
			0.040, 0.108	5.54×10^{-4}	3.0	6.15×10^{-4}	3.0
			0.020, 0.053	4.93×10^{-5}	3.4	5.21×10^{-5}	3.5
			0.010, 0.026	3.63×10^{-6}	3.7	3.77×10^{-6}	3.7

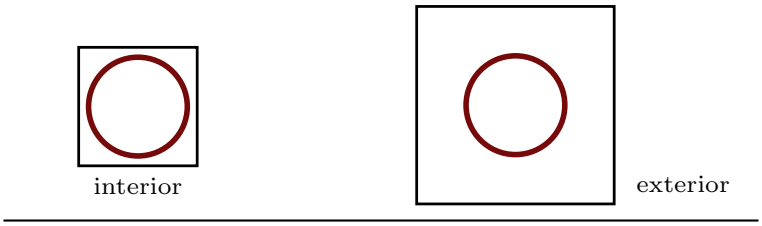
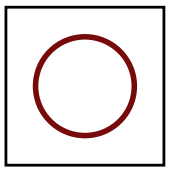


Table 5.5: Interface condition errors for quadratic-log test problem with $\hat{\Omega}^i = [-0.6, -0.6] \times [0.6, 0.6]$.

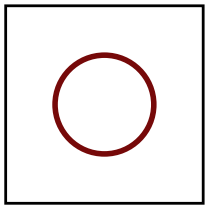
5.3 Numerical solution considerations

As for the exterior case, we consider two options for solving the coupled four-variable linear system: either solve the full system together, or implement the split solve described in (5.23–5.24). As before, the total system is preconditioned by using a block preconditioner for the FE blocks; the preconditioner used in the following tests is again smoothed aggregation AMG, which is known to perform well in these scenarios. For the split solve, the action of inverting the FE operators will again use a reordering to separate the Dirichlet nodes and regain a symmetric positive definite system for the interior points. The total computational

p	qbx	condition	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\partial\Omega)}$	order	$\|\text{error}\|_{L^2(\partial\Omega)}$	order
1	2	value	0.080, 0.224	6.14×10^{-3}	–	5.89×10^{-3}	–
			0.040, 0.108	1.03×10^{-3}	2.5	8.67×10^{-4}	2.6
			0.020, 0.053	2.15×10^{-4}	2.2	1.49×10^{-4}	2.5
			0.010, 0.026	4.98×10^{-5}	2.1	2.97×10^{-5}	2.3
		derivative	0.080, 0.224	8.09×10^{-2}	–	8.28×10^{-2}	–
			0.040, 0.108	5.57×10^{-2}	0.5	3.57×10^{-2}	1.2
			0.020, 0.053	2.80×10^{-2}	1.0	1.67×10^{-2}	1.1
			0.010, 0.026	1.30×10^{-2}	1.1	8.35×10^{-3}	1.0
2	3	value	0.080, 0.224	9.72×10^{-4}	–	1.05×10^{-3}	–
			0.040, 0.108	8.95×10^{-5}	3.3	9.93×10^{-5}	3.2
			0.020, 0.053	6.83×10^{-6}	3.6	7.85×10^{-6}	3.6
			0.010, 0.026	4.63×10^{-7}	3.8	5.55×10^{-7}	3.8
		derivative	0.080, 0.224	8.18×10^{-3}	–	7.64×10^{-3}	–
			0.040, 0.108	1.69×10^{-3}	2.2	1.68×10^{-3}	2.1
			0.020, 0.053	2.86×10^{-4}	2.5	2.73×10^{-4}	2.6
			0.010, 0.026	4.54×10^{-5}	2.6	3.95×10^{-5}	2.8
3	4	value	0.080, 0.224	2.42×10^{-4}	–	2.68×10^{-4}	–
			0.040, 0.108	1.28×10^{-5}	4.0	1.45×10^{-5}	4.0
			0.020, 0.053	5.20×10^{-7}	4.5	6.07×10^{-7}	4.5
			0.010, 0.026	1.91×10^{-8}	4.7	2.21×10^{-8}	4.7
		derivative	0.080, 0.224	2.98×10^{-3}	–	3.24×10^{-3}	–
			0.040, 0.108	3.18×10^{-4}	3.1	3.48×10^{-4}	3.1
			0.020, 0.053	2.71×10^{-5}	3.5	2.93×10^{-5}	3.5
			0.010, 0.026	1.95×10^{-6}	3.8	2.12×10^{-6}	3.7



interior



exterior

Table 5.6: Interface condition errors for quadratic-log test problem with $\hat{\Omega}^i = [-0.8, -0.8] \times [0.8, 0.8]$.

work in the split solve depends on both the number of outer GMRES iterations for the densities and inner iterations on the FE solutions, called every time we need the action of $\underline{\underline{\mathcal{F}}}^{-1}$ as part of the outer operator.

For the sine-linear test case with a circular interface, Figure 5.7 shows the amount of GMRES(20) iterations for the total system solve versus the number of outer GMRES iterations and *total* CG iterations (for every time the inverse function was called). For all iterative solves, a tolerance of 1×10^{-12} was used.

As with the exterior case, the benefit of the split solve technique is not clear, although CG does require less memory than GMRES. However, for the quadratic-log test case, shown in Figure 5.8, the performance is different. As the mesh sizes are decreased, the conditioning of the total system matrix deteriorates

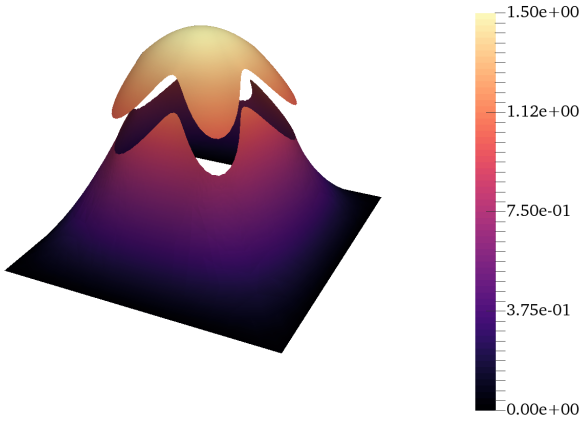
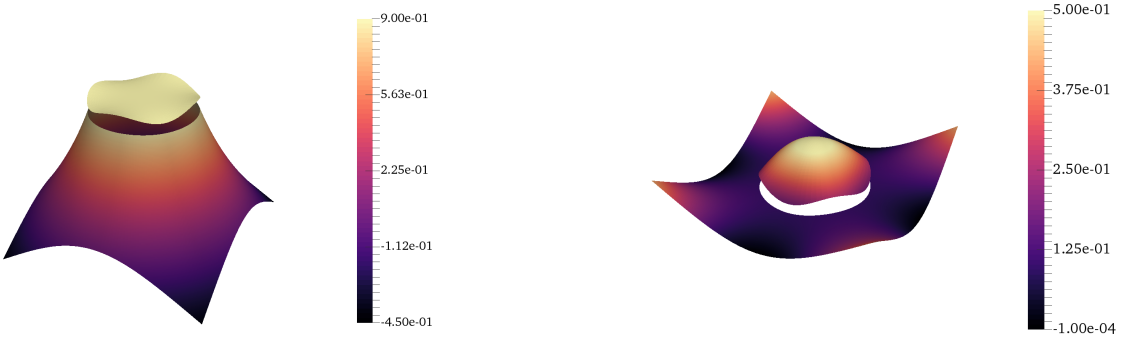


Figure 5.4: Total solution for the "general jump" test case with Γ = starfish (4.51).

	interior	exterior	interface conditions
exact solution	— N/A —		$\kappa = 1/2$
coefficient, β	2	1	$c = 1$
r.h.s, f	10	0	$a(x) = 1/4$
boundary cond.	N/A	0	$b(x) = 0$

Table 5.7: Description of "general jump" case.



(a) IE solution

(b) FE solution

Figure 5.5: FE and IE components of the general jump case with a circular Γ.

convergence well before reaching the solve tolerance, even with the block preconditioner applied. However, the number of outer iterations required by the split solve technique remains more or less constant across all orders of basis functions and all problem sizes. Finally, we compare the effects of $\kappa \neq c$ or $\kappa = c$ on the number of outer iterations required for the split solve in Figure 5.9. The benefit of being able to reduce $\underline{\underline{\mathcal{A}}}$ to the identity matrix when $\kappa = c$ is immediately obvious, as the highest number of outer iterations for all sizes and both domain definitions is only *four*. When $\kappa \neq c$, on the other hand, the number of iterations jumps

p	qbx	condition	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\partial\Omega)}$	order	$\|\text{error}\|_{L^2(\partial\Omega)}$	order
1	2	value	0.080, 0.224	9.27×10^{-3}	–	8.16×10^{-3}	–
			0.040, 0.108	1.47×10^{-3}	2.5	1.43×10^{-3}	2.4
			0.020, 0.053	2.95×10^{-4}	2.3	2.32×10^{-4}	2.6
			0.010, 0.026	6.36×10^{-5}	2.2	4.09×10^{-5}	2.5
		derivative	0.080, 0.224	1.43×10^{-1}	–	1.07×10^{-1}	–
			0.040, 0.108	9.69×10^{-2}	0.5	5.89×10^{-2}	0.8
			0.020, 0.053	4.69×10^{-2}	1.0	2.74×10^{-2}	1.1
			0.010, 0.026	2.31×10^{-2}	1.0	1.38×10^{-2}	1.0
2	3	value	0.080, 0.224	1.63×10^{-3}	–	1.89×10^{-3}	–
			0.040, 0.108	1.59×10^{-4}	3.2	1.80×10^{-4}	3.2
			0.020, 0.053	1.32×10^{-5}	3.5	1.43×10^{-5}	3.6
			0.010, 0.026	9.42×10^{-7}	3.8	1.01×10^{-6}	3.8
		derivative	0.080, 0.224	1.57×10^{-2}	–	1.60×10^{-2}	–
			0.040, 0.108	3.78×10^{-3}	2.0	3.76×10^{-3}	2.0
			0.020, 0.053	6.63×10^{-4}	2.4	6.23×10^{-4}	2.5
			0.010, 0.026	1.07×10^{-4}	2.6	9.11×10^{-5}	2.7
3	4	value	0.080, 0.224	4.32×10^{-4}	–	4.86×10^{-4}	–
			0.040, 0.108	2.39×10^{-5}	4.0	2.65×10^{-5}	4.0
			0.020, 0.053	1.03×10^{-6}	4.4	1.11×10^{-6}	4.5
			0.010, 0.026	3.92×10^{-8}	4.7	4.04×10^{-8}	4.7
		derivative	0.080, 0.224	6.60×10^{-3}	–	7.47×10^{-3}	–
			0.040, 0.108	7.43×10^{-4}	3.0	8.15×10^{-4}	3.0
			0.020, 0.053	6.65×10^{-5}	3.4	6.92×10^{-5}	3.5
			0.010, 0.026	5.05×10^{-6}	3.7	5.02×10^{-6}	3.7

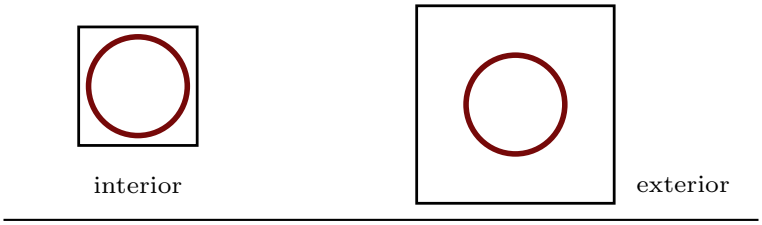
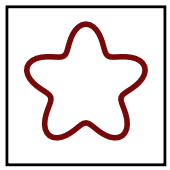


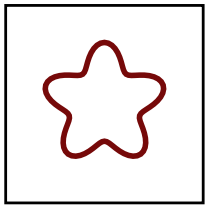
Table 5.8: Interface condition errors for general jump test problem with Ω^i = circle with radius 0.5.

by an order of magnitude. We also see that the interface shape has a much larger impact on the required iterations when $\kappa \neq c$.

p	qbx	condition	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\Omega)}$	order	$\|\text{error}\|_{L^2(\Omega)}$	order
1	2	value	0.080, 0.327	1.84×10^{-2}	–	1.30×10^{-2}	–
			0.040, 0.170	8.50×10^{-3}	1.1	4.47×10^{-3}	1.5
			0.020, 0.085	1.74×10^{-3}	2.3	9.69×10^{-4}	2.2
			0.010, 0.043	2.95×10^{-4}	2.5	1.69×10^{-4}	2.5
		derivative	0.080, 0.327	2.63×10^{-1}	–	2.33×10^{-1}	–
			0.040, 0.170	2.30×10^{-1}	0.2	1.35×10^{-1}	0.8
			0.020, 0.085	1.31×10^{-1}	0.8	6.24×10^{-2}	1.1
			0.010, 0.043	5.56×10^{-2}	1.2	2.60×10^{-2}	1.3
2	3	value	0.080, 0.327	1.16×10^{-2}	–	7.46×10^{-3}	–
			0.040, 0.170	3.92×10^{-3}	1.6	1.65×10^{-3}	2.2
			0.020, 0.085	5.69×10^{-4}	2.8	2.27×10^{-4}	2.9
			0.010, 0.043	4.11×10^{-5}	3.8	2.15×10^{-5}	3.4
		derivative	0.080, 0.327	3.03×10^{-1}	–	1.73×10^{-1}	–
			0.040, 0.170	1.55×10^{-1}	1.0	7.30×10^{-2}	1.2
			0.020, 0.085	6.13×10^{-2}	1.3	2.36×10^{-2}	1.6
			0.010, 0.043	1.60×10^{-2}	1.9	5.62×10^{-3}	2.1
3	4	value	0.080, 0.327	1.22×10^{-2}	–	6.60×10^{-3}	–
			0.040, 0.170	1.50×10^{-3}	3.0	6.97×10^{-4}	3.2
			0.020, 0.085	1.66×10^{-4}	3.2	6.28×10^{-5}	3.5
			0.010, 0.043	7.36×10^{-6}	4.5	3.50×10^{-6}	4.1
		derivative	0.080, 0.327	3.80×10^{-1}	–	2.00×10^{-1}	–
			0.040, 0.170	1.09×10^{-1}	1.8	4.78×10^{-2}	2.1
			0.020, 0.085	3.06×10^{-2}	1.8	1.01×10^{-2}	2.2
			0.010, 0.043	4.96×10^{-3}	2.6	1.45×10^{-3}	2.8



interior



exterior

Table 5.9: Interface condition errors for general jump test problem with Ω^i = starfish curve (4.51).

	interior	exterior	interface conditions
exact solution	$\sin(2\pi x)\sin(2\pi y) + x + y$	$\sin(2\pi x)\sin(2\pi y)$	$\kappa = 1$
coefficient, β	1	1	$c = 1$
r.h.s, f	$4\pi^2\sin(2\pi x)\sin(2\pi y)$	$4\pi^2\sin(2\pi x)\sin(2\pi y)$	$a(x) = x + y$
boundary cond.	*N/A*	*exact soln*	$b(x) = [1, 1]\cdot\mathbf{n}$

Table 5.10: Description of "sine-linear" case.

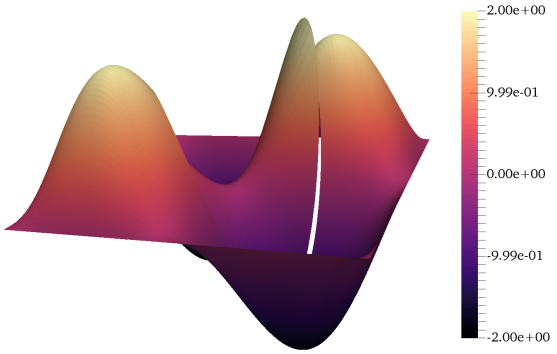


Figure 5.6: Total solution for the "sine-linear" test case.

p	qbx	solution	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\Omega)}$	order	$\|\text{error}\|_{L^2(\Omega)}$	order
1	2	interior	0.080, 0.224	1.21×10^{-2}	–	2.82×10^{-3}	–
			0.040, 0.108	3.67×10^{-3}	1.6	6.34×10^{-4}	2.0
			0.020, 0.053	1.02×10^{-3}	1.8	1.68×10^{-4}	1.9
			0.010, 0.026	3.02×10^{-4}	1.7	4.78×10^{-5}	1.8
1	2	exterior	0.080, 0.224	1.79×10^{-2}	–	1.06×10^{-2}	–
			0.040, 0.108	4.66×10^{-3}	1.8	2.55×10^{-3}	2.0
			0.020, 0.053	1.06×10^{-3}	2.1	6.21×10^{-4}	2.0
			0.010, 0.026	3.14×10^{-4}	1.7	1.70×10^{-4}	1.8
2	3	interior	0.080, 0.224	1.38×10^{-3}	–	2.83×10^{-4}	–
			0.040, 0.108	1.33×10^{-4}	3.2	2.38×10^{-5}	3.4
			0.020, 0.053	1.47×10^{-5}	3.1	2.38×10^{-6}	3.2
			0.010, 0.026	2.24×10^{-6}	2.7	3.56×10^{-7}	2.7
2	3	exterior	0.080, 0.224	4.64×10^{-4}	–	2.66×10^{-4}	–
			0.040, 0.108	8.65×10^{-5}	2.3	3.36×10^{-5}	2.8
			0.020, 0.053	8.81×10^{-6}	3.2	3.71×10^{-6}	3.1
			0.010, 0.026	1.42×10^{-6}	2.6	5.91×10^{-7}	2.6
3	4	interior	0.080, 0.224	1.62×10^{-4}	–	3.10×10^{-5}	–
			0.040, 0.108	8.60×10^{-6}	4.0	1.01×10^{-6}	4.7
			0.020, 0.053	4.47×10^{-7}	4.2	9.49×10^{-8}	3.3
			0.010, 0.026	2.90×10^{-8}	3.9	3.40×10^{-9}	4.7
3	4	exterior	0.080, 0.224	3.76×10^{-5}	–	9.80×10^{-6}	–
			0.040, 0.108	4.27×10^{-6}	3.0	1.25×10^{-6}	2.8
			0.020, 0.053	4.21×10^{-7}	3.3	1.24×10^{-7}	3.3
			0.010, 0.026	2.51×10^{-8}	4.0	4.35×10^{-9}	4.8

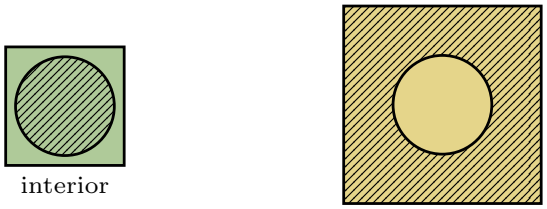
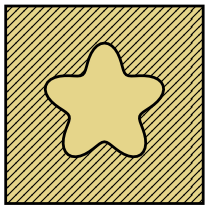


Table 5.11: Error convergence for sine-linear test problem with Ω^i = circle of radius 0.5.

p	qbx	solution	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\Omega)}$	order	$\|\text{error}\|_{L^2(\Omega)}$	order
1	2	interior	0.080, 0.327	7.43×10^{-2}	–	1.66×10^{-2}	–
			0.040, 0.170	5.54×10^{-3}	3.7	1.27×10^{-3}	3.7
			0.020, 0.085	1.37×10^{-3}	2.0	3.17×10^{-4}	2.0
			0.010, 0.043	3.63×10^{-4}	1.9	7.87×10^{-5}	2.0
1	2	exterior	0.080, 0.327	7.32×10^{-2}	–	2.63×10^{-2}	–
			0.040, 0.170	5.63×10^{-3}	3.7	3.08×10^{-3}	3.1
			0.020, 0.085	1.45×10^{-3}	2.0	7.28×10^{-4}	2.1
			0.010, 0.043	3.48×10^{-4}	2.0	1.93×10^{-4}	1.9
2	3	interior	0.080, 0.327	2.11×10^{-3}	–	3.91×10^{-4}	–
			0.040, 0.170	3.41×10^{-4}	2.6	2.64×10^{-5}	3.9
			0.020, 0.085	1.76×10^{-5}	4.3	2.69×10^{-6}	3.3
			0.010, 0.043	2.90×10^{-6}	2.6	6.97×10^{-7}	1.9
2	3	exterior	0.080, 0.327	3.01×10^{-3}	–	4.55×10^{-4}	–
			0.040, 0.170	1.29×10^{-4}	4.5	3.26×10^{-5}	3.8
			0.020, 0.085	1.12×10^{-5}	3.5	3.98×10^{-6}	3.0
			0.010, 0.043	2.87×10^{-6}	2.0	8.99×10^{-7}	2.1
3	4	interior	0.080, 0.327	2.92×10^{-3}	–	2.74×10^{-4}	–
			0.040, 0.170	4.78×10^{-4}	2.6	1.21×10^{-5}	4.5
			0.020, 0.085	1.78×10^{-5}	4.7	3.33×10^{-7}	5.2
			0.010, 0.043	1.25×10^{-6}	3.8	1.04×10^{-8}	5.0
3	4	exterior	0.080, 0.327	3.79×10^{-3}	–	3.44×10^{-4}	–
			0.040, 0.170	1.88×10^{-4}	4.3	9.34×10^{-6}	5.2
			0.020, 0.085	1.30×10^{-5}	3.9	2.43×10^{-7}	5.3
			0.010, 0.043	5.24×10^{-7}	4.6	7.44×10^{-9}	5.0

interior

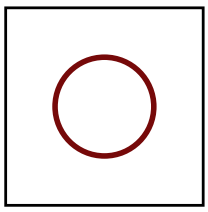


exterior

Table 5.12: Error convergence for sine-linear test problem with Ω^i = starfish curve (4.51).

p	qbx	condition	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\partial\Omega)}$	order	$\|\text{error}\|_{L^2(\partial\Omega)}$	order
1	2	value	0.080, 0.224	7.78×10^{-3}	–	7.17×10^{-3}	–
			0.040, 0.108	1.59×10^{-3}	2.2	1.09×10^{-3}	2.6
			0.020, 0.053	2.85×10^{-4}	2.4	1.71×10^{-4}	2.6
			0.010, 0.026	4.93×10^{-5}	2.5	3.13×10^{-5}	2.4
		derivative	0.080, 0.224	2.79×10^{-1}	–	2.25×10^{-1}	–
			0.040, 0.108	1.36×10^{-1}	1.0	8.21×10^{-2}	1.4
			0.020, 0.053	6.66×10^{-2}	1.0	3.63×10^{-2}	1.1
			0.010, 0.026	2.37×10^{-2}	1.5	1.54×10^{-2}	1.2
2	3	value	0.080, 0.224	1.21×10^{-3}	–	1.32×10^{-3}	–
			0.040, 0.108	1.06×10^{-4}	3.3	9.15×10^{-5}	3.7
			0.020, 0.053	8.29×10^{-6}	3.6	6.34×10^{-6}	3.8
			0.010, 0.026	6.61×10^{-7}	3.6	4.80×10^{-7}	3.7
		derivative	0.080, 0.224	5.33×10^{-2}	–	4.98×10^{-2}	–
			0.040, 0.108	7.73×10^{-3}	2.6	6.70×10^{-3}	2.8
			0.020, 0.053	1.54×10^{-3}	2.3	1.03×10^{-3}	2.6
			0.010, 0.026	2.97×10^{-4}	2.3	1.68×10^{-4}	2.6
3	4	value	0.080, 0.224	1.46×10^{-4}	–	1.44×10^{-4}	–
			0.040, 0.108	4.75×10^{-6}	4.7	4.37×10^{-6}	4.8
			0.020, 0.053	1.67×10^{-7}	4.7	1.36×10^{-7}	4.9
			0.010, 0.026	7.11×10^{-9}	4.5	4.25×10^{-9}	4.9
		derivative	0.080, 0.224	6.23×10^{-3}	–	6.33×10^{-3}	–
			0.040, 0.108	5.68×10^{-4}	3.3	4.08×10^{-4}	3.8
			0.020, 0.053	3.06×10^{-5}	4.1	2.24×10^{-5}	4.1
			0.010, 0.026	3.48×10^{-6}	3.1	1.68×10^{-6}	3.7

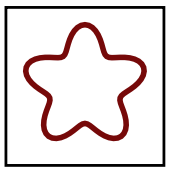
interior



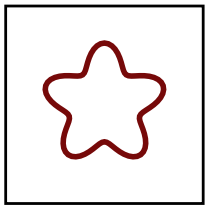
exterior

Table 5.13: Interface condition error for sine-linear test problem with Ω^i = circle of radius 0.5.

p	qbx	condition	h_{fe}, h_{ie}	$\|\text{error}\|_{L^\infty(\partial\Omega)}$	order	$\|\text{error}\|_{L^2(\partial\Omega)}$	order
1	2	value	0.080, 0.327	3.02×10^{-2}	–	1.72×10^{-2}	–
			0.040, 0.170	1.11×10^{-3}	4.8	9.42×10^{-4}	4.2
			0.020, 0.085	1.97×10^{-4}	2.5	1.46×10^{-4}	2.7
			0.010, 0.043	3.73×10^{-5}	2.4	2.24×10^{-5}	2.7
		derivative	0.080, 0.327	$1.23\times10^{+0}$	–	7.76×10^{-1}	–
			0.040, 0.170	7.15×10^{-2}	4.1	4.52×10^{-2}	4.1
			0.020, 0.085	3.64×10^{-2}	1.0	1.63×10^{-2}	1.5
			0.010, 0.043	1.62×10^{-2}	1.2	7.50×10^{-3}	1.1
2	3	value	0.080, 0.327	4.62×10^{-3}	–	2.29×10^{-3}	–
			0.040, 0.170	3.06×10^{-4}	3.9	1.34×10^{-4}	4.1
			0.020, 0.085	9.75×10^{-6}	5.0	8.06×10^{-6}	4.1
			0.010, 0.043	6.66×10^{-7}	3.9	5.19×10^{-7}	3.9
		derivative	0.080, 0.327	8.80×10^{-2}	–	5.58×10^{-2}	–
			0.040, 0.170	1.28×10^{-2}	2.8	7.28×10^{-3}	2.9
			0.020, 0.085	2.05×10^{-3}	2.6	9.29×10^{-4}	3.0
			0.010, 0.043	2.39×10^{-4}	3.1	1.17×10^{-4}	3.0
3	4	value	0.080, 0.327	5.75×10^{-3}	–	2.74×10^{-3}	–
			0.040, 0.170	4.47×10^{-4}	3.7	1.10×10^{-4}	4.6
			0.020, 0.085	1.82×10^{-5}	4.6	3.78×10^{-6}	4.9
			0.010, 0.043	6.15×10^{-7}	4.9	1.10×10^{-7}	5.1
		derivative	0.080, 0.327	1.61×10^{-1}	–	9.57×10^{-2}	–
			0.040, 0.170	2.99×10^{-2}	2.4	1.04×10^{-2}	3.2
			0.020, 0.085	4.54×10^{-3}	2.7	1.09×10^{-3}	3.2
			0.010, 0.043	5.08×10^{-4}	3.1	8.86×10^{-5}	3.6

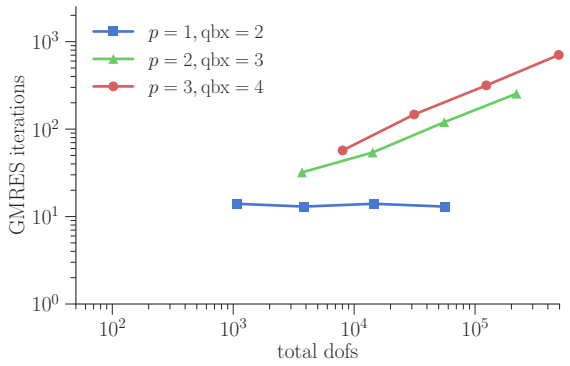


interior

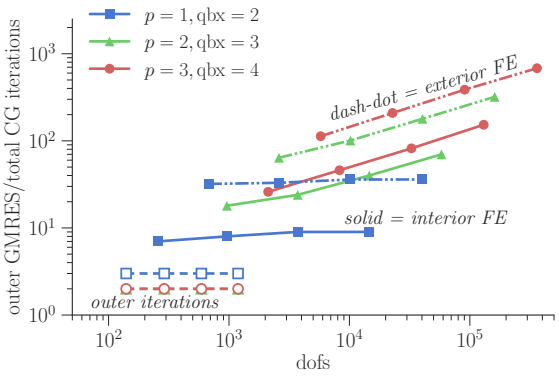


exterior

Table 5.14: Interface condition error for sine-linear test problem with Ω^i = starfish curve (4.51).

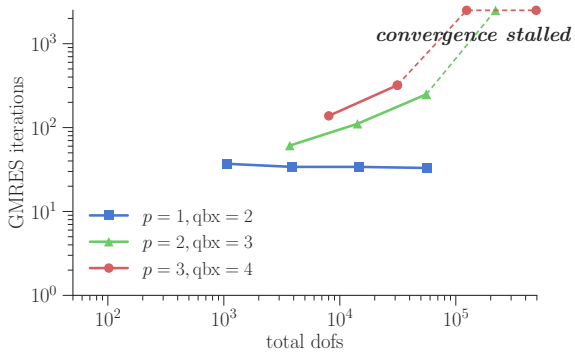


(a) Iterations of preconditioned GMRES(20) for the total coupled system.

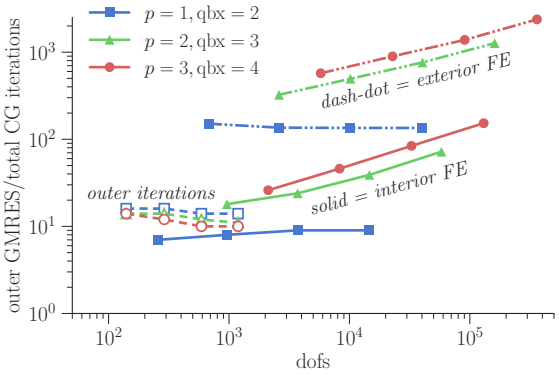


(b) Outer GMRES(20) and inner interior/exterior preconditioned CG iterations for the split solve.

Figure 5.7: Comparison of solution techniques for the sine-linear test problem with Ω^i = circle of radius 0.5.

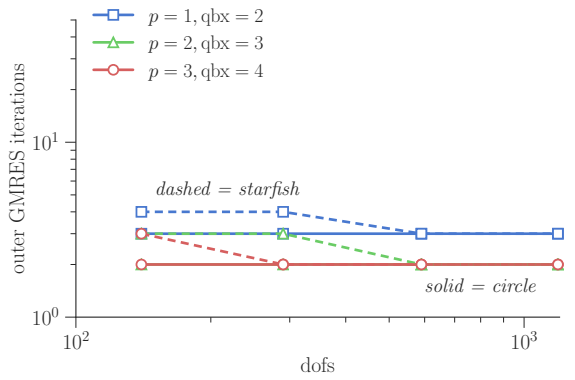


(a) Iterations of preconditioned GMRES(20) for the total coupled system.

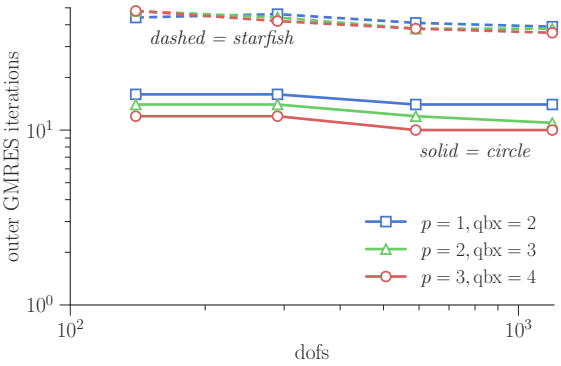


(b) Outer GMRES(20) and inner interior/exterior preconditioned CG iterations for the split solve.

Figure 5.8: Comparison of solution techniques for the quadratic-log test problem with Ω^i = circle of radius 0.5.



(a) Outer iterations for sine-linear case, with $\kappa = c$.



(b) Outer iterations for general jump case, with $\kappa = 0.5$ and $c = 1$.

Figure 5.9: Effects of $\kappa = c$ or $\kappa \neq c$ on outer iterations for split solve.

Chapter 6

FE-IE for Stokes flow

Stokes flow describes the flow regime where viscous forces greatly outweigh inertial forces — i.e., when the Reynolds number $\ll 1$. In this case, the full steady-state incompressible Navier-Stokes equations

$$\rho \mathbf{u} \cdot \nabla \mathbf{u} = \mu \nabla \cdot \nabla \mathbf{u} - \nabla p + \mathbf{f} \tag{6.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{6.2}$$

for velocity $\mathbf{u}$, pressure p, viscosity μ, density ρ, and body force $\mathbf{f}$, reduce to the Stokes equations

$$\mu \nabla \cdot \nabla \mathbf{u} - \nabla p + \mathbf{f} = 0 \tag{6.3}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{6.4}$$

Stokes flow is applicable in highly viscous fluids, at very low velocities, or in flows with very small characteristic length scales. Without inertia, the effects of applying a force are felt instantaneously throughout the domain, a key feature of elliptic equations but perhaps not always intuitive when considering fluid flow.

The Stokes equations are linear, which makes them suitable for Green's function solutions. Indeed, there are hydrodynamic analogues to the Laplace single- and double-layer potentials. Excellent overviews of hydrodynamic layer potentials and their use in representing general Stokes flows are given in [64] and [65]. We will briefly summarize some key definitions and features of these layer potentials in the next section.

6.1 Layer Potentials for Stokes flow

The single-layer potential for Stokes flow is defined

$$\mathcal{V}\boldsymbol{\gamma} = \int_{\partial\Omega} G_{ij}(\mathbf{x}, \mathbf{x}_0)\gamma_j(\mathbf{x}_0)\, d\mathbf{x}_0. \tag{6.5}$$

The Green's function is the "Stokeslet," and when applied to a directional vector, it represents the flow due to a point force in that direction. It is a dyadic Green's function — a tensor of rank two — which links the components of the vector density so that the Stokeslet can automatically satisfy the incompressibility condition. A velocity given by a single layer paired with its associated pressure field, given by

$$p_{\mathcal{V}} = \int_{\partial\Omega} \phi_i(\mathbf{x}, \mathbf{x}_0)\gamma_i(\mathbf{x}_0)\, d\mathbf{x}_0, \tag{6.6}$$

satisfies the homogeneous incompressible Stokes equations. The Stokeslet and its pressure kernel $\mathbf{\Phi}$ are shown in the first two rows of Table 6.1 for two and three dimensions (here $\hat{\mathbf{x}}$ represents the vector $\mathbf{x} - \mathbf{x}_0$). Like the Laplace single layer $\mathcal{S}$, $\mathcal{V}$ is continuous at the boundary $\partial\Omega$. As implied by the fact that its kernel is the gradient of the Laplace Green's function, the Stokeslet pressure has a jump at the boundary.

kernel	$d = 2$	$d = 3$
Stokeslet, G_{ij}	$-\frac{1}{4\pi\mu}\left(-\delta_{ij}\log r + \frac{\hat{x}_i\hat{x}_j}{r^2}\right)$	$-\frac{1}{8\pi\mu}\left(\delta_{ij}\frac{1}{r} + \frac{\hat{x}_i\hat{x}_j}{r^3}\right)$
pressure, ϕ_i	$-\frac{1}{2\pi}\frac{\hat{x}_i}{r^2} = \frac{\partial}{\partial x_i}\left(-\frac{1}{2\pi}\log r\right)$	$-\frac{1}{4\pi}\frac{\hat{x}_i}{r^3} = \frac{\partial}{\partial x_i}\left(\frac{1}{4\pi}\frac{1}{r}\right)$
stresslet, T_{ijk}	$\frac{1}{\pi}\frac{\hat{x}_i\hat{x}_j\hat{x}_k}{r^4}$	$\frac{3}{4\pi}\frac{\hat{x}_i\hat{x}_j\hat{x}_k}{r^5}$
pressure, Π_{ij}	$\frac{\mu}{\pi}\left(-\frac{\delta_{ij}}{r^2} + 2\frac{\hat{x}_i\hat{x}_j}{r^4}\right) = 2\mu\frac{\partial}{\partial x_j}\frac{\partial}{\partial x_i}\left(-\frac{1}{2\pi}\log r\right)$	$\frac{\mu}{2\pi}\left(-\frac{\delta_{ij}}{r^3} + 3\frac{\hat{x}_i\hat{x}_j}{r^5}\right) = 2\mu\frac{\partial}{\partial x_j}\frac{\partial}{\partial x_i}\left(\frac{1}{4\pi}\frac{1}{r}\right)$

Table 6.1: Kernels for Stokes layer potentials.

The double-layer potential for Stokes flow is not directly built from the normal derivative of the single layer, like $\mathcal{D}$. Rather, its kernel is the "stresslet," so named because it represents the viscous stress tensor related to the Stokeslet. That is, with the viscous stress tensor defined as

$$\sigma_{ik} = -p\delta_{ik} + \mu\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right), \tag{6.7}$$

the stresslet $\mathbf{T}$ is

$$T_{ijk} = -\phi_j\delta_{ik} + \mu\left(\frac{\partial G_{ij}}{\partial x_k} + \frac{\partial G_{kj}}{\partial x_i}\right). \tag{6.8}$$

The double layer is then

$$\mathcal{W} = \int_{\partial\Omega} T_{ijk}(\mathbf{x}, \mathbf{x}_0)n_k(\mathbf{x}_0)\gamma_j(\mathbf{x}_0)\, d\mathbf{x}_0, \tag{6.9}$$

with $\mathbf{n}$ as the outward unit normal vector. The pressure of the double layer is

$$p_{\mathcal{W}} = \int_{\partial\Omega} \Pi_{ik}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}_0) \gamma_i(\mathbf{x}_0)\, d\mathbf{x}_0. \tag{6.10}$$

Like $\mathcal{D}$, $\mathcal{W}$ has a jump equal to the density $\boldsymbol{\gamma}$ at the boundary $\partial\Omega$. With the sign convention in Table 6.1, we have

$$\begin{aligned} \mathcal{W}_-\boldsymbol{\gamma} &= -\frac{1}{2}\boldsymbol{\gamma} + (\mathcal{W})\boldsymbol{\gamma} \\ \mathcal{W}_+\boldsymbol{\gamma} &= \quad \frac{1}{2}\boldsymbol{\gamma} + (\mathcal{W})\boldsymbol{\gamma}. \end{aligned} \tag{6.11}$$

Thus the double layer potential would, as with the Laplace equation, result in second-kind integral equations for the Dirichlet problem. As suggested by the relationship between the stresslet and the Stokeslet, the single layer potential has a jump in the normal stress, and gives second-kind equations for Neumann problems related to the traction $\boldsymbol{\sigma} \cdot \mathbf{n}$.

6.2 FE-IE for Stokes flow around objects

In this section, we demonstrate how the exterior FE-IE splitting can be extended to solve Stokes flow around rigid bodies, as shown in Figure 6.1.

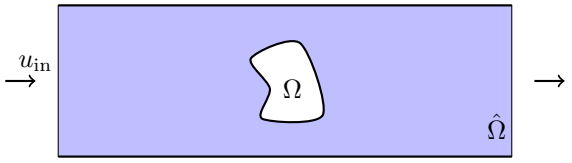


Figure 6.1: Exterior problem of Stokes flow inside $\hat{\Omega}$ and outside submerged rigid body Ω.

As a sample problem, we consider the case where we wish to enforce Dirichlet boundary conditions on the fluid velocity at the body's surface, and either Dirichlet or Neumann (based on the traction, $\boldsymbol{\sigma} \cdot \hat{n}$) conditions on the faces of the domain $\hat{\Omega}$:

$$
\begin{aligned}
\mu\nabla\cdot\nabla\mathbf{u}-\nabla p+\mathbf{f}&=0 \qquad x\in\hat{\Omega}\backslash\Omega\\
\nabla\cdot\mathbf{u}&=0\\
\mathbf{u}&=\mathbf{v} \qquad x\in\partial\Omega\\
\mathbf{u}&=\hat{\mathbf{v}}_D \qquad x\in\partial\hat{\Omega}_D\\
\boldsymbol{\sigma}\cdot\mathbf{n}&=\hat{\mathbf{t}}_N \qquad x\in\partial\hat{\Omega}_N \quad (\text{where } \partial\hat{\Omega}_N\cup\partial\hat{\Omega}_D=\partial\hat{\Omega}).
\end{aligned}
\tag{6.12}
$$

FE-IE splitting

Following the idea of Section 4.3, we split the problem into an interior problem in $\hat{\Omega}$ for the FE solver and a purely exterior problem in $\mathbb{R}^d$ for the IE solver. Again, the FE solver handles any forces $\mathbf{f}$. Thus (6.12) becomes

$$
\begin{aligned}
\text{[FE]}\quad \mu\nabla\cdot\nabla\mathbf{u}_1-\nabla p_1+\mathbf{f}&=0 \qquad x\in\hat{\Omega}\\
\nabla\cdot\mathbf{u}_1&=0\\
\mathbf{u}_1&=\hat{\mathbf{v}}_D-\hat{\mathbf{r}}_2 \qquad x\in\partial\hat{\Omega}_D\\
\boldsymbol{\sigma}_1\cdot\mathbf{n}&=\hat{\mathbf{t}}_N-\hat{\mathbf{t}}_2 \qquad x\in\partial\hat{\Omega}_N
\end{aligned}
\tag{6.13}
$$

$$
\begin{aligned}
\text{[IE]}\quad \mu\nabla\cdot\nabla\mathbf{u}_2-\nabla p_2&=0 \qquad x\in\mathbb{R}^2\backslash\Omega\\
\nabla\cdot\mathbf{u}_2&=0\\
\mathbf{u}_2&=\mathbf{v}-\mathbf{r}_1 \qquad x\in\partial\Omega,
\end{aligned}
\tag{6.14}
$$

where $\hat{\mathbf{r}}_2$ is the restriction of $\mathbf{u}_2$ to $\partial\hat{\Omega}_D$, $\hat{\mathbf{t}}_2$ is the restriction of the traction $\boldsymbol{\sigma}_2\cdot\mathbf{n}$ to $\partial\hat{\Omega}_N$, and $\mathbf{r}_1$ is velocity $\mathbf{u}_1$ on the rigid body surface $\partial\Omega$.

Layer potential representation

As mentioned previously, the double layer $\mathcal{W}$ has a jump across the boundary and will result in a second-kind integral equation for the Dirichlet boundary problem. However, in (6.14), we want to solve an exterior equation. The double layer on its own is not enough to represent any exterior Stokes flow; its nullspace is of dimension $3(d-1)$ where d is the spatial dimension and $d=2$ or 3. This nullspace is spanned by the

rigid body modes of motion — d translation modes and $3(d-1)-d$ rotation modes [64]. There are several previously proposed approaches to modifying the kernel by adding extra terms in order to represent these modes. Similar to Section 4.3, we take the combined-single-and-double-layer approach and choose

$$\mathbf{u}_2 = \mathcal{W}\boldsymbol{\gamma} + \mathcal{V}\boldsymbol{\gamma}, \tag{6.15}$$

which for (6.14) results in the integral equation

$$\left\{\frac{1}{2}I + (\mathcal{W}) + (\mathcal{V})\right\}\boldsymbol{\gamma} = \mathbf{v} - \mathbf{r}_1. \tag{6.16}$$

In three dimensions, Hebeker [66] showed that this integral equation has a unique solution. In two dimensions, the logarithmic behavior of the single layer complicates the issue. Hsiao and Kress [67] use a representation with a double layer, a single layer with a constant density scaled match the logarithmic boundary condition at infinity, a single layer with the mean value of the density removed (which can represent rotational modes, but will not affect a non-zero logarithmic boundary condition at infinity), and the integral of the density (which adds translational modes). In our case, however, we do not need to explicitly match any logarithmic boundary conditions at infinity, as the solution beyond $\hat{\Omega}$ is immaterial. The careful combination of a single layer with no net force and a single layer with a specific net force — as we can think of the two single-layer terms in the Hsiao-Kress representation — is not necessary. Furthermore, after coupling with the FE solver we expect that adding the constant translational modes is not only no longer required, but detrimental, as seen in the exterior Laplace case: constant velocities are also in the range of the FE solver, and thus adding it to the IE operator would mean our FE-IE decomposition would not be unique. Therefore, we use the simple and general combination representation (6.15) for both two- and three-dimensional problems. The pressure field corresponding to $\mathbf{u}_2$ is then computed by

$$p_2 = \int_{\partial\Omega} \phi_i(\mathbf{x}, \mathbf{x}_0)\gamma_i(\mathbf{x}_0) + \Pi_{ik}(\mathbf{x}, \mathbf{x}_0)n_k(\mathbf{x}_0)\gamma_i(\mathbf{x}_0)\, d\mathbf{x}_0. \tag{6.17}$$

Finite element formulation

We use a standard mixed finite element formulation for the Stokes equations, which results in a symmetric saddle point problem for u_1 and p_1.

First, we note that

$$\begin{aligned}\nabla \cdot \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) &= \partial x_j \frac{\partial u_i}{\partial x_j} + \partial x_j \frac{\partial u_j}{\partial x_i} \\ &= \frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{\partial^2 u_j}{\partial x_j \partial x_i} \\ &= \nabla \cdot \nabla \mathbf{u} + \nabla(\nabla \cdot \mathbf{u}).\end{aligned}$$

Since we want $\mathbf{u}$ to be incompressible and will be solving $\nabla \cdot \mathbf{u} = 0$, we arrive at

$$\nabla \cdot \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) = \nabla \cdot \nabla \mathbf{u}. \tag{6.18}$$

Returning to (6.13), we use the substitution (6.18) and multiply the equations by -1 to find

$$\begin{aligned}-\mu \nabla \cdot \left(\nabla \mathbf{u}_1 + \nabla \mathbf{u}_1^T\right) + \nabla p_1 - \mathbf{f} &= 0 \\ -\nabla \cdot \mathbf{u}_1 &= 0.\end{aligned} \tag{6.19}$$

These seemingly cosmetic changes to the incompressible Stokes equations are done so that the resulting bilinear form is symmetric and the natural boundary conditions are related to the traction on $\partial\hat{\Omega}_N$, as we shall see.

We introduce the test function pair $(\mathbf{w}, q) \in \mathbf{H}^1(\hat{\Omega}) \times L^2(\hat{\Omega})$:

$$\int_{\hat{\Omega}} -\mu \left(\nabla \mathbf{u}_1 + \nabla \mathbf{u}_1^T\right) \cdot \mathbf{w} \, dV + \int_{\hat{\Omega}} \nabla p_1 \cdot \mathbf{w} \, dV - \int_{\hat{\Omega}} (\nabla \cdot \mathbf{u}_1) q \, dV = \int_{\hat{\Omega}} \mathbf{f} \cdot \mathbf{w} \, dV. \tag{6.20}$$

Integrating by parts yields

$$\begin{aligned}\int_{\partial\hat{\Omega}} -\mu \mathbf{w} \cdot \left(\nabla \mathbf{u}_1 + \nabla \mathbf{u}_1^T\right) \cdot \hat{n} \, dS + \int_{\hat{\Omega}} \mu \left(\nabla \mathbf{u}_1 + \nabla \mathbf{u}_1^T\right) : \nabla \mathbf{w} \, dV + \int_{\partial\hat{\Omega}} \mathbf{w} \cdot (p_1 \mathbf{I} \cdot \hat{n}) \, dS - \int_{\hat{\Omega}} p_1 (\nabla \cdot \mathbf{w}) \, dV \\ - \int_{\hat{\Omega}} (\nabla \cdot \mathbf{u}_1) q \, dV = \int_{\hat{\Omega}} \mathbf{f} \cdot \mathbf{w} \, dV.\end{aligned} \tag{6.21}$$

Or, grouping the surface terms, we have

$$\begin{aligned}\int_{\partial\hat{\Omega}} \mathbf{w} \cdot \left(p_1 \mathbf{I} - \mu \left(\nabla \mathbf{u}_1 + \nabla \mathbf{u}_1^T\right)\right) \cdot \hat{n} \, dS + \int_{\hat{\Omega}} \mu \left(\nabla \mathbf{u}_1 + \nabla \mathbf{u}_1^T\right) : \nabla \mathbf{w} \, dV - \int_{\hat{\Omega}} p_1 (\nabla \cdot \mathbf{w}) \, dV \\ - \int_{\hat{\Omega}} (\nabla \cdot \mathbf{u}_1) q \, dV = \int_{\hat{\Omega}} \mathbf{f} \cdot \mathbf{w} \, dV.\end{aligned} \tag{6.22}$$

Finally, we notice that $p_1\mathbf{I} - \mu\left(\nabla\mathbf{u}_1 + \nabla\mathbf{u}_1^T\right) = -\boldsymbol{\sigma}\cdot\hat{n}$ and $\left(\nabla\mathbf{u}_1 + \nabla\mathbf{u}_1^T\right) : \nabla\mathbf{w} = \left(\nabla\mathbf{u}_1 + \nabla\mathbf{u}_1^T\right) : \frac{1}{2}\left(\nabla\mathbf{w} + \nabla\mathbf{w}^T\right)$, and arrive at the final weak form before enforcing boundary conditions:

Find $(\mathbf{u}_1, p_1) \in \mathbf{H}^1(\hat{\Omega}) \times L^2(\hat{\Omega})$ *such that*

$$\begin{aligned}\int_{\hat{\Omega}} \frac{\mu}{2}\left(\nabla\mathbf{u}_1 + \nabla\mathbf{u}_1^T\right) : \left(\nabla\mathbf{w} + \nabla\mathbf{w}^T\right)\, dV - \int_{\hat{\Omega}} p_1(\nabla\cdot\mathbf{w})\, dV - \int_{\hat{\Omega}} (\nabla\cdot\mathbf{u}_1) q\, dV \\ = \int_{\partial\hat{\Omega}} \mathbf{w}\cdot(\boldsymbol{\sigma}_1\cdot\hat{n})\, dS + \int_{\hat{\Omega}} \mathbf{f}\cdot\mathbf{w}\, dV \end{aligned} \tag{6.23}$$

$\forall(\mathbf{w}, q) \in \mathbf{H}^1(\hat{\Omega}) \times L^2(\hat{\Omega})$.

We define the operators $\mathcal{F}(\mathbf{w})$, $\mathcal{B}(q)$, and $\mathcal{B}^T(\mathbf{w})$ such that

$$\mathcal{F}(\mathbf{w})\mathbf{u} = \int_{\hat{\Omega}} \frac{\mu}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right) : \left(\nabla\mathbf{w} + \nabla\mathbf{w}^T\right)\, dV, \tag{6.24}$$

$$\mathcal{B}(q)\mathbf{u} = -\int_{\hat{\Omega}} q(\nabla\cdot\mathbf{u})\, dV, \tag{6.25}$$

and

$$\mathcal{B}^T(\mathbf{w})p = -\int_{\hat{\Omega}} p(\nabla\cdot\mathbf{w})\, dV. \tag{6.26}$$

Also, similar to the definition in Section 4.3, consider

$$\mathcal{M}(\mathbf{w})\mathbf{f} = \int_{\hat{\Omega}} \mathbf{f}\cdot\mathbf{w}\, dV. \tag{6.27}$$

Before enforcing boundary conditions on the system, the result is

$$\left[\begin{array}{c|c} \mathcal{F}(\mathbf{w}) & \mathcal{B}^T(\mathbf{w}) \\ \hline \mathcal{B}(q) & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{u}_1 \\ \hline p_1 \end{array}\right] = \left[\begin{array}{c} \mathcal{M}(\mathbf{w})\,\mathbf{f} \\ \hline 0 \end{array}\right] \tag{6.28}$$

which is symmetric (but indefinite) for a Galerkin formulation using the same space for trial and test basis functions.

To enforce any Neumann boundary conditions, we add the surface integral term over the Neumann portion of the boundary:

$$\mathcal{M}(\mathbf{w};\hat{\mathbf{t}})\,\mathbf{f} = \int_{\hat{\Omega}} \mathbf{f}\cdot\mathbf{w}\, dV + \int_{\partial\hat{\Omega}_N} \hat{\mathbf{t}}\cdot\mathbf{w}\, dS. \tag{6.29}$$

We will require that the test functions $\mathbf{w} \in \mathbf{H}^1(\hat{\Omega})$ be equal to zero on the Dirichlet boundary $\partial\hat{\Omega}_D$, removing the rest of the surface integral from (6.23). This, along with our non-standard way of applying essential boundary conditions, also means modifying the operators $\mathcal{F}$ and $\mathcal{M}$, similarly to Section 4.3. Again, we

denote the modified operators as $\hat{\mathcal{F}}(\mathbf{w})$ and $\hat{\mathcal{M}}(\mathbf{w};\hat{\mathbf{t}}_N;\hat{\mathbf{v}}_D)$. Now we also need to modify $\mathcal{B}^T(\mathbf{w})$, simply by removing the contributions from the affected test functions on the Dirichlet boundary. We will call the new operator $\mathcal{B}^T_{\text{int}}(\mathbf{w})$. Armed with these definitions, we write the terms in the weak form (6.23) with boundary conditions for the FE portion of the splitting (6.13) as:

$$\left[\begin{array}{c|c} \hat{\mathcal{F}}(\mathbf{w}) & \mathcal{B}^T_{\text{int}}(\mathbf{w}) \\ \hline \mathcal{B}(q) & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{u}_1 \\ \hline p_1 \end{array}\right] = \left[\begin{array}{c} \hat{\mathcal{M}}(\mathbf{w};\hat{\mathbf{t}}_N - \hat{\mathbf{t}}_2;\hat{\mathbf{v}}_D)\,\mathbf{f} \\ \hline 0 \end{array}\right]. \tag{6.30}$$

In the case of Dirichlet boundary conditions on the pressure p_1, $\mathcal{B}(q)$ is modified in a similar way to $\mathcal{B}^T_{\text{int}}(\mathbf{w})$, and the appropriate entries would be added in the currently-empty bottom-right quadrant of the operator on $[\mathbf{u}_1; p_1]$.

We note that in addition to the coupling operators $\mathcal{R}$ and $\hat{\mathcal{R}}$, we require an operator to handle the fact that $\hat{\mathbf{t}}_2$ appears in (6.30) but depends on the vector density $\boldsymbol{\gamma}$ and is also unknown. We denote this operator $\hat{\mathcal{Q}}(\mathbf{w})$ (due to its association with quadrature in its discrete form, as we shall see later) and define it as follows: the operator $\hat{\mathcal{Q}}(\mathbf{w})$ for a layer potential representation $\mathcal{A}$ (and its associated pressure kernel) takes $\boldsymbol{\gamma}$ and returns the surface integral

$$\hat{\mathcal{Q}}(\mathbf{w}) = \int_{\partial\hat{\Omega}_N} (\boldsymbol{\sigma}_{\mathcal{A}} \cdot \mathbf{n}) \cdot \mathbf{w}\, dS, \tag{6.31}$$

for any $\mathbf{w} \in \mathbf{H}^1(\hat{\Omega})$, where $\boldsymbol{\sigma}_{\mathcal{A}}$ is the viscous stress tensor for the Stokes flow $\mathcal{A}\boldsymbol{\gamma}$. Now we can write the coupled FE-IE problem as:

Find $(\boldsymbol{\gamma}, \mathbf{u}_1, p_1)$ *such that*

$$\left[\begin{array}{c|cc} \frac{1}{2}I + (\mathcal{W}) + (\mathcal{V}) & \mathcal{R} & 0 \\ \hline \hat{\mathcal{R}}^D + \hat{\mathcal{Q}}(\mathbf{w}) & \hat{\mathcal{F}}(\mathbf{w}) & \mathcal{B}^T_{\text{int}}(\mathbf{w}) \\ 0 & \mathcal{B}(q) & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\gamma} \\ \hline \mathbf{u}_1 \\ p_1 \end{array}\right] = \left[\begin{array}{c} \mathbf{v} \\ \hline \hat{\mathcal{M}}(\mathbf{w};\hat{\mathbf{t}}_N;\hat{\mathbf{v}}_D)\,\mathbf{f} \\ 0 \end{array}\right] \tag{6.32}$$

$\forall (\mathbf{w}, q) \in \mathbf{H}^1(\hat{\Omega}) \times L^2(\hat{\Omega})$. Here $\hat{\mathcal{R}}^D$ indicates that the operator is only applied to the Dirichlet boundary $\partial\hat{\Omega}_D$.

Practical numerical computation

In this section, we will discuss specifics of how to compute the discrete operators associated with the continuous problem (6.32).

First, we denote the matrices arising from the discretization of the single- and double-layer boundary operators as (V) and (W), respectively. R and $\hat{R}^D$ are constructed in the same way as in Section 4.3,

except that the same process for building the matrices is repeated for each component of the velocity and $\hat{R}^D$ indicates that it only has non-zero rows for the nodes associated with the Dirichlet boundary. Again, we impose essential boundary conditions in a different manner than the standard FE approach. $F_{\text{int}} + \hat{I}^D$ represents the finite element matrix with Dirichlet-boundary-associated rows zeroed and added to the identity matrix for these nodes, while $\tilde{\mathbf{f}}_{\text{int}}$ is the discrete form of $\mathcal{M}(\mathbf{w})\,\mathbf{f}$ with Dirichlet-associated entries removed.

For the Neumann boundary, we need a discretized version of the $\hat{\mathcal{Q}}$ operator. First, we choose a quadrature rule for each element to approximate integration on the Neumann element faces. Define $\hat{Q}_k$ as a sparse $N \times N_q$ matrix where N is the total number of degrees of freedom in the FE problem and N_q is the total number of quadrature points on the Neumann faces. Each non-zero row for $\hat{Q}_k$ corresponds to a degree of freedom on the Neumann boundary for the k component of velocity, and

$$\hat{Q}_{k_{ij}} = w_i(\mathbf{x}_j)\vartheta_j,$$

where $w_i(\mathbf{x}_j)$ is the ith basis function evaluated at the jth face quadrature point, ϑ_j is the appropriate quadrature weighting for $\mathbf{x}_j$, and we have restricted ourselves to $i \in \mathcal{N}_k$ where $\mathcal{N}_k$ is the set of all degrees of freedom associated with both the Neumann boundary and the kth component of velocity. Then, with $\mathbf{e}_k$ as the kth unit direction vector, we define

$$\hat{t}^q_{k_j} = \hat{\mathbf{t}}_N(\mathbf{x}_j) \cdot \mathbf{e}_k \quad \forall \quad \mathbf{x}_j \text{ in face quadrature rule},$$

which is the kth component of the traction evaluated at the quadrature points. Thus, the known part of the Neumann boundary integral, which appears in the right hand side of (6.32) as part of $\hat{\mathcal{M}}(\mathbf{w}; \hat{\mathbf{t}}_N; \hat{\mathbf{v}}_D)\,\mathbf{f}$, can be approximated as $\sum_{k=1}^{d} \hat{Q}_k \hat{t}^q_k$. The final result corresponds to the usual way of approximating surface integrals for Neumann boundary conditions through quadrature, but we use the notation for consistency with the operator for the unknown part of the condition $\hat{\mathbf{t}}_2$.

For $\hat{\mathbf{t}}_2$, we also introduce the matrices $\hat{R}^N_k$, defined as follows: Let $\hat{R}^N_k$ be the $N_q \times m$ matrix whose rows represent the viscous stress of the discretized layer potential representation, contracted with the outward normal vector and evaluated at the face quadrature points for $\hat{Q}_k$. Or rather, we build this matrix by column, with δ_j as defined in Section 4.3 — a vector of length m which is equal to one for the jth element and zero everywhere else. Then the columns of $\hat{R}^N_k$ are $(\boldsymbol{\sigma}^h_2 \cdot \mathbf{n}) \cdot \mathbf{e}_k \delta_j$, evaluated at the quadrature points. Then, the vector result of $\sum_{k=1}^{d} \hat{Q}_k \hat{R}^N_k \boldsymbol{\gamma}^h$ approximates the surface integral of the traction $\hat{\mathbf{t}}_2$ multiplied by the basis function for each Neumann node of $\mathbf{U}_1$, and is zero for all other degrees of freedom.

In total, the discrete system is

$$\left[\begin{array}{c|cc} \frac{1}{2}I + (W) + (V) & R & 0 \\ \hline \hat{R}^D + \sum_{k=1}^d \hat{Q}_k \hat{R}_k^N & F_{\text{int}} + \hat{I}^D & B_{\text{int}}^T \\ 0 & B & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\gamma}^h \\ \hline \mathbf{U}_1 \\ P_1 \end{array}\right] = \left[\begin{array}{c} \mathbf{v} \\ \hline \tilde{\mathbf{f}}_{\text{int}} + \sum_{k=1}^d \hat{Q}_k \hat{t}_k^q + \hat{I}^D \hat{\mathbf{v}}_D \\ 0 \end{array}\right]. \tag{6.33}$$

Here $\boldsymbol{\gamma}^h$ is the discrete vector density and $\mathbf{U}_1$ and P_1 are the coefficients of the FE solution at the mesh points.

Nested solution process

Unfortunately, as with the Poisson equation, our non-standard way of enforcing Dirichlet boundary conditions ruins the symmetry of the FE matrix. However, as before, we recover the symmetry of the velocity-velocity matrix by performing a nested iteration based on a split solve. That is,

1. Solve

$$\left\{\frac{1}{2}I + (W) + (V) - \left[\begin{array}{cc} R & 0 \end{array}\right] \left[\begin{array}{cc} F_{\text{int}} + \hat{I}^D & B_{\text{int}}^T \\ B & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \hat{R}^D + \sum_{k=1}^d \hat{Q}_k \hat{R}_k^N \\ 0 \end{array}\right]\right\} \boldsymbol{\gamma}^h =$$
$$\mathbf{v} - \left[\begin{array}{cc} R & 0 \end{array}\right] \left[\begin{array}{cc} F_{\text{int}} + \hat{I}^D & B_{\text{int}}^T \\ B & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \tilde{\mathbf{f}}_{\text{int}} + \sum_{k=1}^d \hat{Q}_k \hat{t}_k^q + \hat{I}^D \hat{\mathbf{v}}_D \\ 0 \end{array}\right], \tag{6.34}$$

   then

2. Solve

$$\left[\begin{array}{c} \mathbf{U}_1 \\ P_1 \end{array}\right] = \left[\begin{array}{cc} F_{\text{int}} + \hat{I}^D & B_{\text{int}}^T \\ B & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \tilde{\mathbf{f}}_{\text{int}} + \sum_{k=1}^d \hat{Q}_k \hat{t}_k^q + \hat{I}^D \hat{\mathbf{v}}_D \\ 0 \end{array}\right]$$
$$- \left[\begin{array}{cc} F_{\text{int}} + \hat{I}^D & B_{\text{int}}^T \\ B & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \hat{R}^D + \sum_{k=1}^d \hat{Q}_k \hat{R}_k^N \\ 0 \end{array}\right] \boldsymbol{\gamma}^h. \tag{6.35}$$

The focus for splitting and symmetrizing is the inverse action

$$\left[\begin{array}{cc} F_{\text{int}} + \hat{I}^D & B_{\text{int}}^T \\ B & 0 \end{array}\right]^{-1}.$$

This is split as

1. Solve

$$\left\{B[F_{\text{int}} + \hat{I}^D]^{-1}B_{\text{int}}^T\right\} P_1 = B[F_{\text{int}} + \hat{I}^D]^{-1}\left(\tilde{\mathbf{f}}_{\text{int}} + \sum_{k=1}^{d} \hat{Q}_k \hat{t}_k^q + \hat{I}^D \hat{\mathbf{v}}_D\right), \tag{6.36}$$

   then

2. Solve

$$\mathbf{U}_1 = [F_{\text{int}} + \hat{I}^D]^{-1}\left(\tilde{\mathbf{f}}_{\text{int}} + \sum_{k=1}^{d} \hat{Q}_k \hat{t}_k^q + \hat{I}^D \hat{\mathbf{v}}_D - B_{\text{int}}^T P_1\right). \tag{6.37}$$

Finally, in order to recover a symmetric matrix for the inner solves involving $[F_{\text{int}} + \hat{I}^D]^{-1}$, we perform the same reordering as before, by placing all Dirichlet nodes at the end. This isolates the symmetric matrix $F_{\text{int-int}}$:

$$\left[\begin{array}{c|c} F_{\text{int}} + \hat{I}^D & B_{\text{int}}^T \\ \hline B & 0 \end{array}\right]\left[\begin{array}{c} \mathbf{U}_1 \\ \hline P_1 \end{array}\right] = \left[\begin{array}{cc|c} F_{\text{int-int}} & F_{\text{int-ext}} & B_{\text{int}}^T \\ 0 & \hat{I}^D & 0 \\ \hline \multicolumn{2}{c|}{B} & 0 \end{array}\right]\left[\begin{array}{c} \mathbf{U}_{1_{\text{int}}} \\ \mathbf{U}_{1_D} \\ \hline P_1 \end{array}\right],$$

where $\mathbf{U}_{1_D}$ are the Dirichlet boundary nodes.

6.3 Numerical tests

Two dimensions

We consider the numerical test of a linear shear flow over a plate and past a stationary cylinder, with no-slip boundary conditions enforced on the cylinder surface. This case has been studied both in theory [68] and experiment [69]. The numerical experiment parameters are described in Table 6.2 and the setup is shown in Figure 6.2. The viscosity was chosen based on the kinematic viscosity of 20 cm/s^2 reported in Taneda's experimental setup with glycerine as the working fluid and assuming an approximate density of 1.26 g/cm^3 [70]. Figure 6.3 shows the FE, IE, and total solutions for the case where the gap between the cylinder and the plate, ϵ, is 0.6. The mesh sizes for this case were $h_{fe} = 0.1$ and $h_{ie} = 0.052$. The standard $Q_2 - Q_1$ finite element pair was used for the velocity and pressure basis function space, which is well-known to be stable for the Stokes problem. The IE solver used third order QBX for the results shown.

Figure 6.4 compares the numerical experiments to the experimental results. We see separation from the plate and the formation of vortices for $\epsilon = 0.25$, as predicted by theory. Furthermore, Figure 6.5 compares the present numerical experiments to theory for an ϵ small enough that theory predicts a small separation

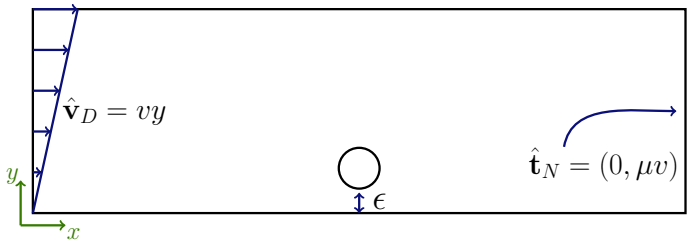


Figure 6.2: Cylinder-plate numerical experiment.

viscosity, μ	25.2 g/cm·s
shear rate, v	0.22 s^{-1}
cylinder radius, R	0.5 cm
domain	$[-8, 0] \times [8, 5]$ cm
Dirichlet faces	$y = 0$, $y = 5$, $x = -8$
Neumann face	$x = 8$

Table 6.2: Parameters for 2D numerical tests.

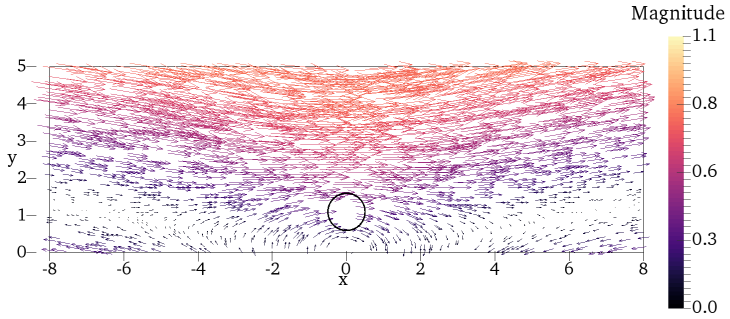
from the cylinder and larger vortices at the plate.

We also consider flow over a starfish cylinder, as given by (4.51). Figure 6.6 shows the FE, IE, and total solutions, and Figure 6.7 shows the formation of vortices in this case as well for $\epsilon = 0.25$.

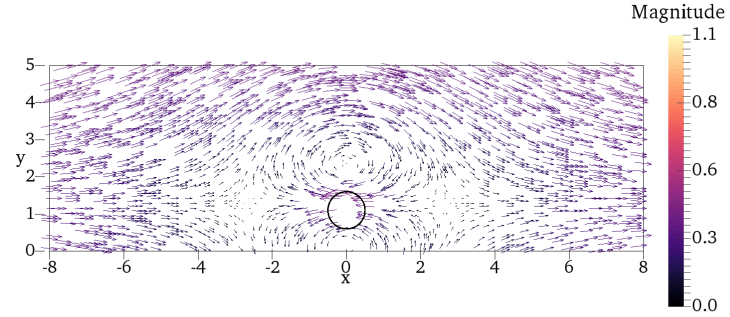
Unfortunately, as ϵ decreases, the outer problem in the nested solve — solving for $\boldsymbol{\gamma}^h$ in (6.34) — becomes increasingly poorly conditioned, as shown in Figure 6.8. The analysis of the outer operator for the exterior problem in Section 4.3 illuminates why this occurs. As ϵ grows small, the difference between the solution to the interior homogeneous problem, represented by $[F_{\text{int}} + \hat{I}^D]^{-1}$, and the exterior solution $(W)\boldsymbol{\gamma}^h + (V)\boldsymbol{\gamma}^h$ will be very small on $\partial\Omega$. This is demonstrated in Figure 6.9, where we see the result of applying the coupled FE-IE operator on $\partial\Omega$ to the constant vector $[\mathbf{1}; \mathbf{1}]$. The x-component of the result is plotted for both $\epsilon = 0.6$ and $\epsilon = 0.01$. The size of the marker indicates the magnitude of the value at that point. The result is that for small ϵ, a large piece of the domain $\partial\Omega$ has its input nearly obliterated in the x direction.

Shear flow past a rotating cylinder

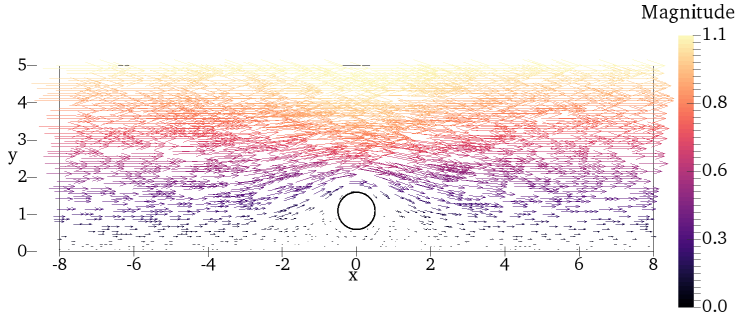
With the FE-IE method, we can easily change the essential boundary conditions on the cylinder. This is demonstrated by replacing $\mathbf{v} = 0$ on the cylinder surface to rigid body rotation, $\mathbf{v} = \omega r$, with r being the distance from the cylinder's axis of rotation. The near-field results for counterclockwise rotation with $\omega = 1$ rad/s are shown for both the circular and starfish cylinders with $\epsilon = 0.6$ in Figure 6.10. We see that by adding the counterclockwise rotation, separation from the plate has been induced for this value of ϵ, whereas we saw no separation in the stationary case. We also see that the vortices form further away from the



(a) FE solution, cm/s.



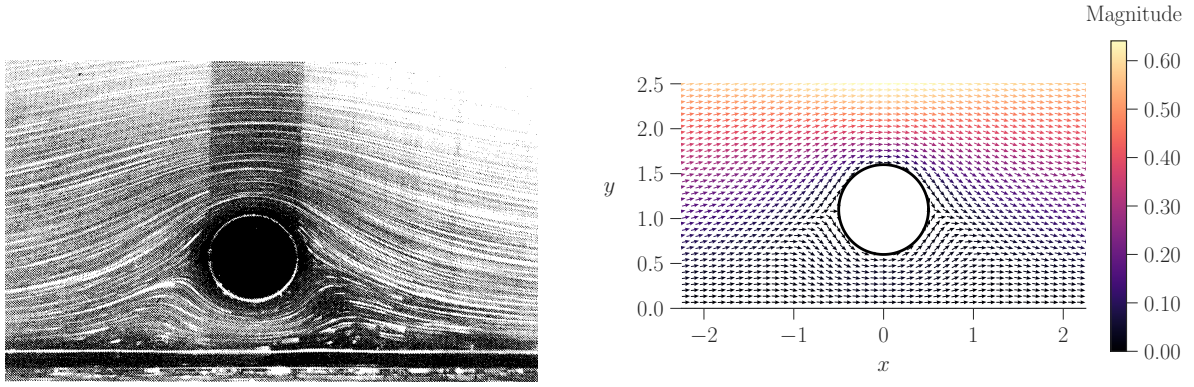
(b) IE solution, cm/s.



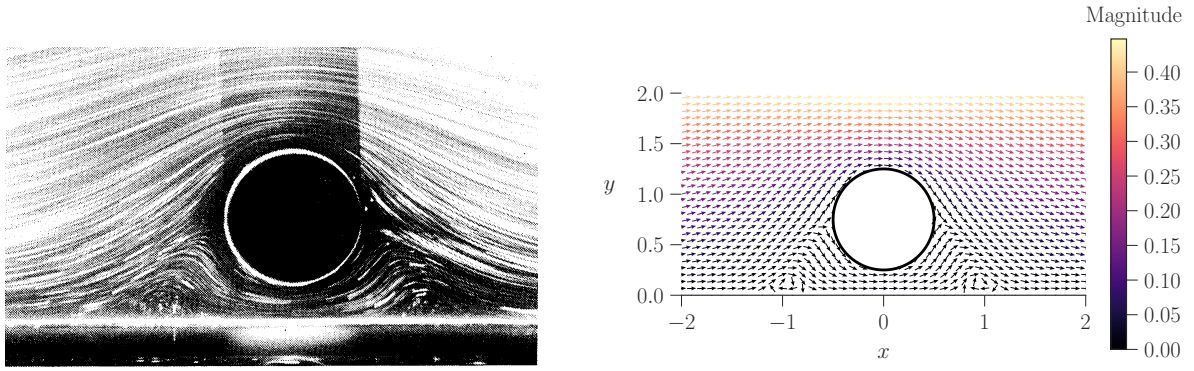
(c) Total solution, cm/s.

Figure 6.3: Shear flow around a rigid circular cylinder.

cylinder along the plate.

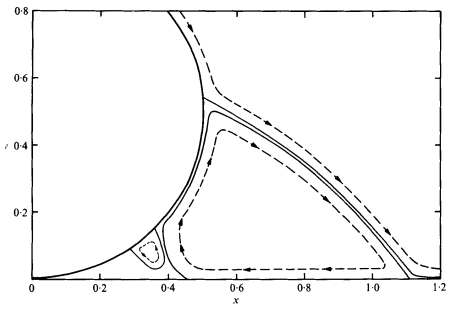


(a) $\epsilon = 0.6$

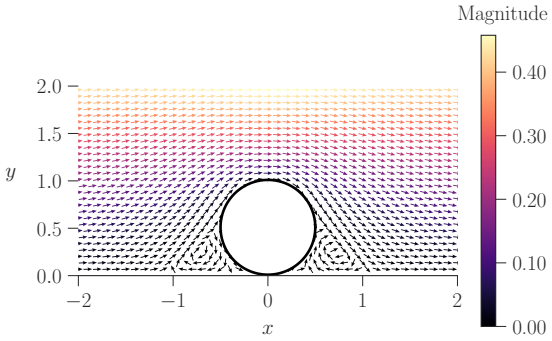


(b) $\epsilon = 0.25$

Figure 6.4: Shear flow over a cylinder: experimental results by Taneda [69] and current numerical tests.

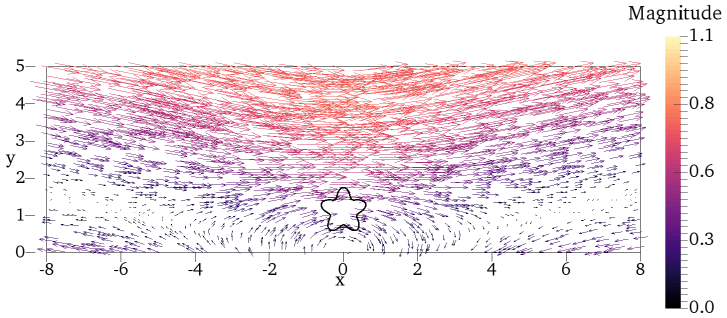


(a) Theoretical results of Davis and O'Neill [68] for a case after separate from the cylinder, with $\epsilon \approx 0.005$.

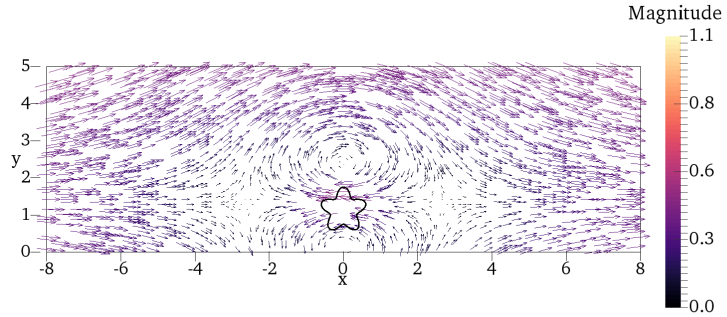


(b) Numerical results with $\epsilon = 0.01$.

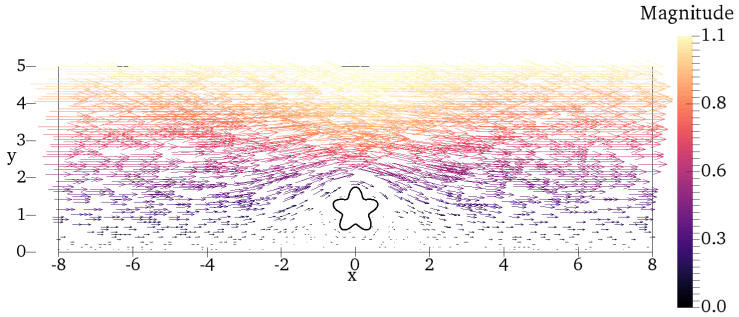
Figure 6.5: Separation from the cylinder does not occur until very small ϵ.



(a) FE solution, cm/s.

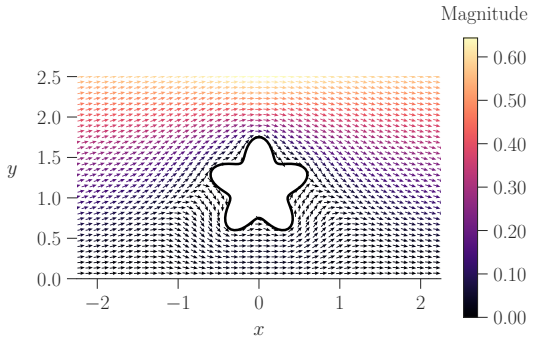


(b) IE solution, cm/s.

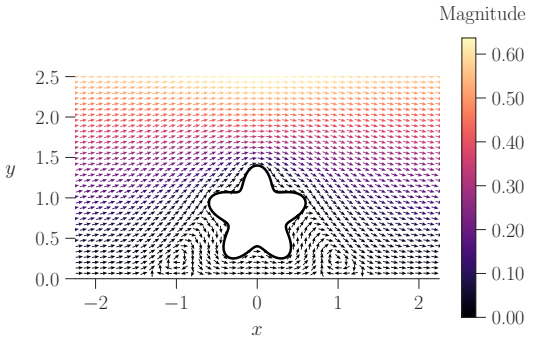


(c) Total solution, cm/s.

Figure 6.6: Shear flow around a rigid starfish (4.51) cylinder.



(a) $\epsilon = 0.6$



(b) $\epsilon = 0.25$

Figure 6.7: Separation from the plate also occurs between $\epsilon = 0.6$ and $\epsilon = 0.25$ for the starfish cylinder.

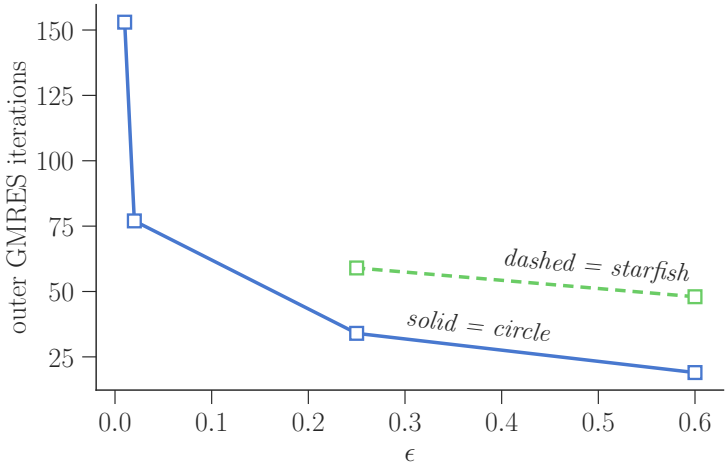
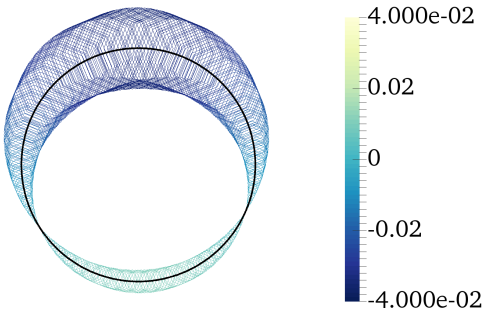
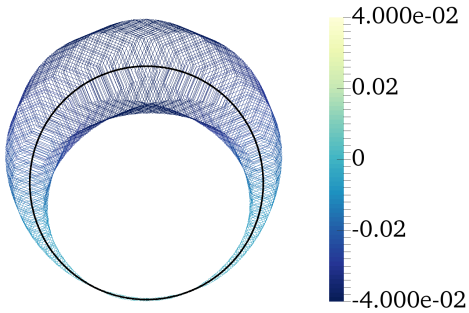


Figure 6.8: Comparison of outer iterations required vs ϵ for the cylinder-plate problem.

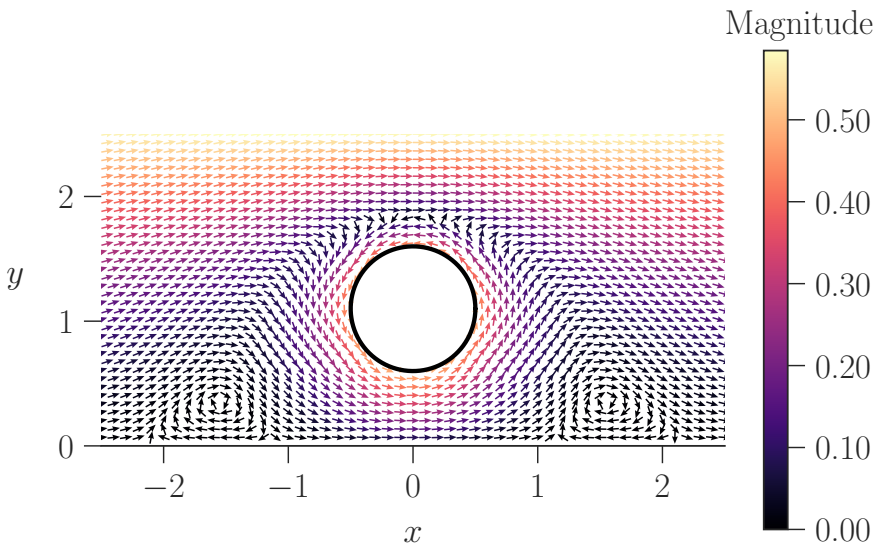


(a) $\epsilon = 0.6$

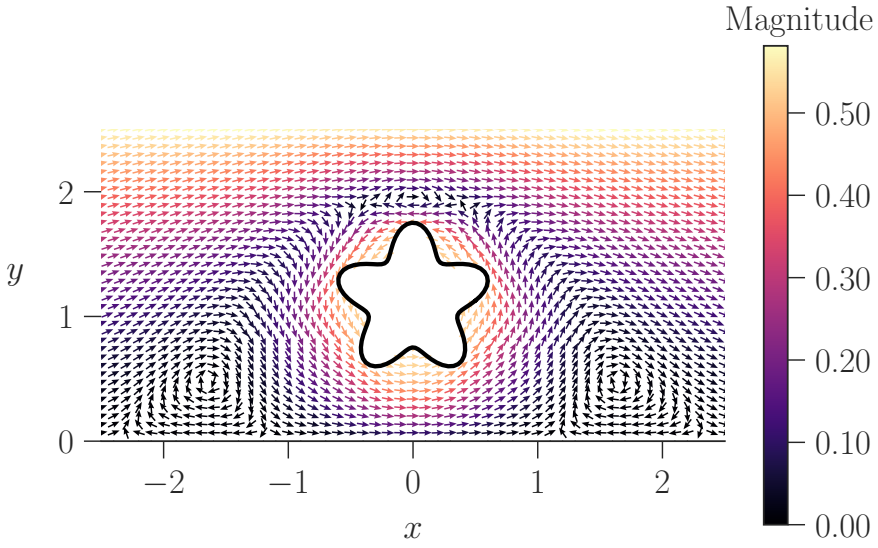


(b) $\epsilon = 0.01$

Figure 6.9: x-component of the result from applying the outer operator for $\boldsymbol{\gamma}$ to the constant vector $[\mathbf{1}; \mathbf{1}]$. The size of the marker indicates the magnitude of the component.



(a) Circular cylinder



(b) Starfish cylinder

Figure 6.10: Effects of counterclockwise rotation of $\omega = 1$ rad/s on the cylinder-plate shear flow problem with $\epsilon = 0.6$.

Three dimensions

In this section, we demonstrate the method in three dimensions. The test case considered is that of uniform flow past a sphere. In the case of free-space boundary conditions — i.e., that the flow matches the uniform condition as distance from the sphere approaches infinity — an exact solution is known, given in spherical coordinates from the center of the sphere as

$$u_r = U\cos\theta\left(1 - \frac{3}{2}\frac{R}{r} + \frac{1}{2}\frac{R^3}{r^3}\right)$$
$$u_\theta = -U\sin\theta\left(1 - \frac{3}{4}\frac{R}{r} - \frac{1}{4}\frac{R^3}{r^3}\right)$$
$$u_\phi = 0 \tag{6.38}$$
$$p = -\frac{3}{2}\mu U\cos\theta\frac{R}{r^2} \tag{6.39}$$

with U as the uniform flow velocity, R as the sphere radius, and θ as the angle measured from the axis along the direction of U. Additionally, the total drag force on the sphere in this flow is given by

$$F_d = \int_{\partial\Omega} \boldsymbol{\sigma}\cdot\mathbf{n}\,dx = 6\pi\mu RU \tag{6.40}$$

in the direction of U, which is the famous Stokes law result.

To verify the three dimensional solver, we first test this exact solution. The test parameters are shown in Figure 6.11 and Table 6.3. For this verification test, Dirichlet boundary conditions were used on all faces, since the exact solution is known everywhere. *Note:* the boundary condition is less than U on all faces, as the exact solution has not reached its asymptotic behavior yet at the edge of the computational domain. With no Neumann faces, we must apply at least one extra condition to the pressure in order to have a unique solution; Dirichlet boundary conditions were used for the pressure as well.

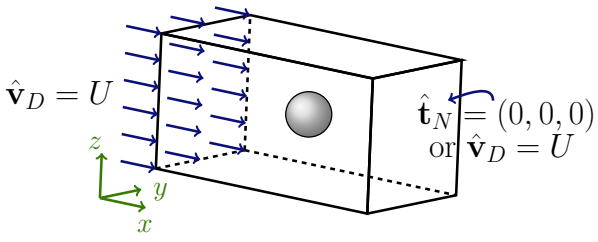


Figure 6.11: Sphere in uniform flow.

The error convergence for two meshes is shown in Table 6.4 for the three components of velocity, the pressure, and the drag force. The FE solver uses the standard Q_2-Q_1 finite element pair for the velocity and

viscosity, μ	25.2 g/cm·s
uniform flow, U	0.5 cm/s
sphere radius, R	0.5 cm
domain	$[-3, -1.5, -1.5] \times [3, 1.5, 1.5]$ cm
Neumann face	$x = 3$ or none

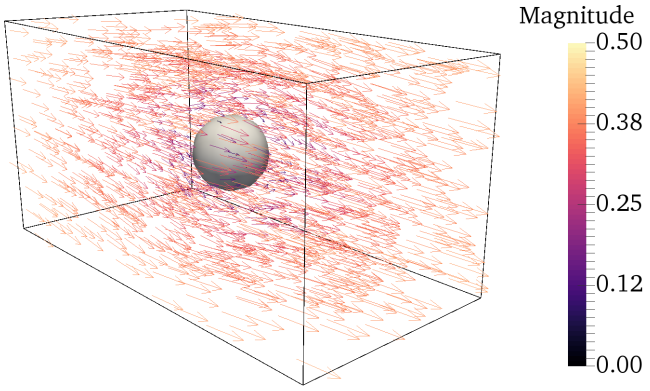
Table 6.3: Parameters for 3D numerical tests.

h_{fe}, h_{ie}	$\|\text{error}_{u_x}\|_{L^\infty}$	$\|\text{error}_{u_y}\|_{L^\infty}$	$\|\text{error}_{u_z}\|_{L^\infty}$	$\|\text{error}_p\|_{L^\infty}$	error_{F_d}
0.5, 0.212	1.74×10^{-3}	1.02×10^{-3}	1.50×10^{-3}	0.595	0.142
0.25, 0.108	8.15×10^{-4}	2.52×10^{-4}	3.36×10^{-4}	0.711	0.208

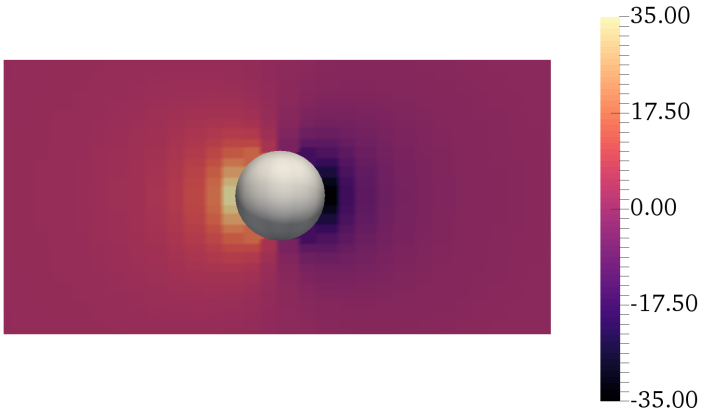
Table 6.4: Errors for unbounded uniform flow around a sphere.

pressure, and the IE solver uses fourth order QBX. The exact drag force for this test case is approximately 118.75 g·cm/s^2 (1.1875×10^{-3} N). Interestingly, while we see convergence in the velocity errors, we do not see the same in the pressure and drag force. The magnitude of the pressure and drag force, however, are much higher than the velocity, so the discrepancy in the size of the relative errors is small. The total velocity field and a slice of the pressure through $z = 0$ are shown in Figure 6.12.

Next we apply the uniform boundary condition on the Dirichlet faces of the computational box in Figure 6.11. The FE, IE, and total velocity fields are shown in Figure 6.13, and slices of the pressure fields are shown in Figure 6.14. The drag force in this case is increased to 282.2 g·cm/s^2 (2.822×10^{-3} N).

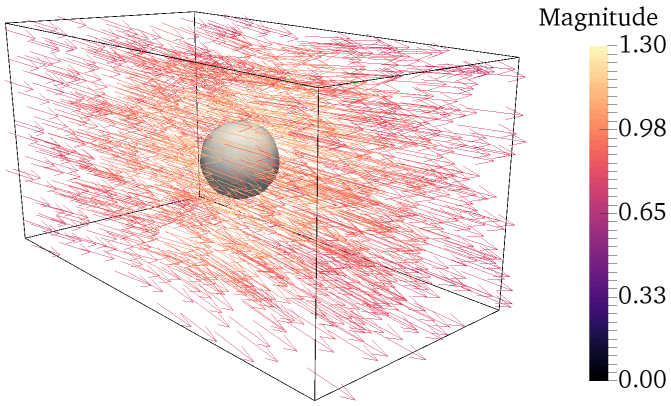


(a) Total velocity field, cm/s.

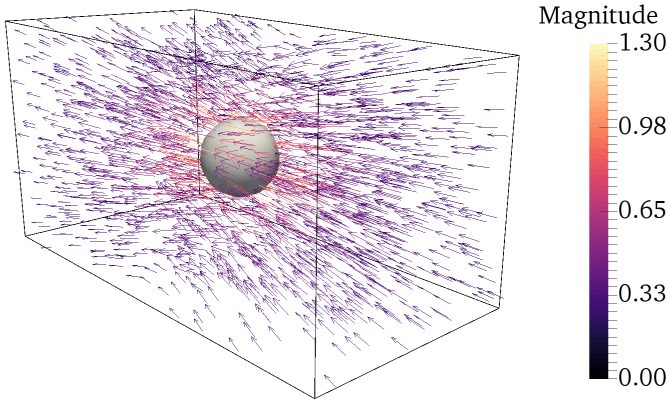


(b) Slice of total pressure field (g/cm·s^2, = 0.1 Pa) at $z = 0$.

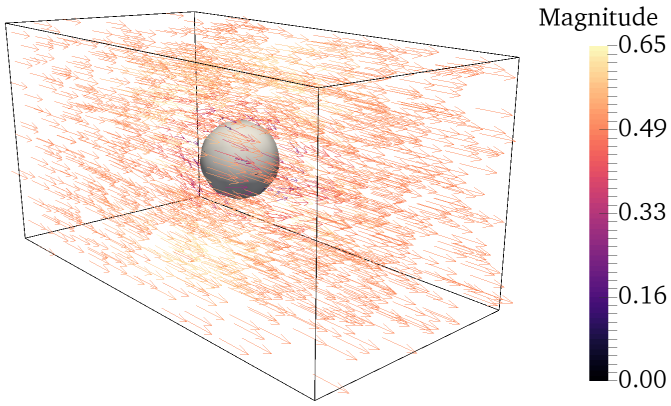
Figure 6.12: Approximation of exact solution for unbounded uniform flow over a sphere.



(a) FE solution, cm/s.

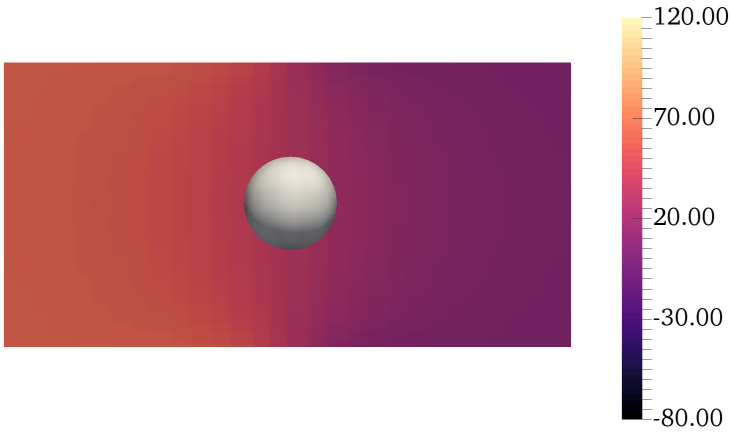


(b) IE solution, cm/s.

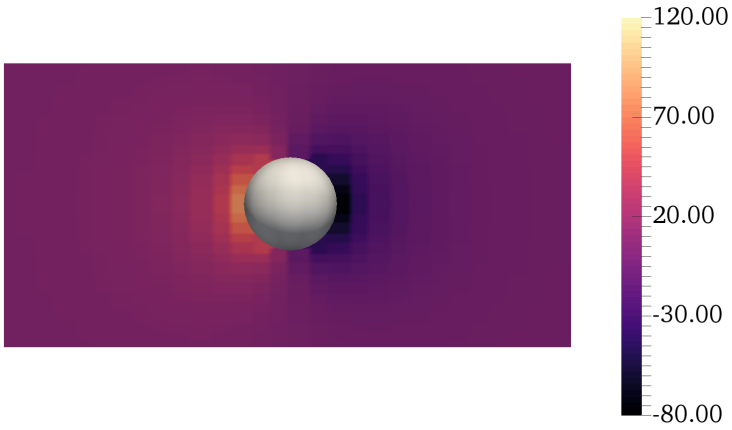


(c) Total solution, cm/s.

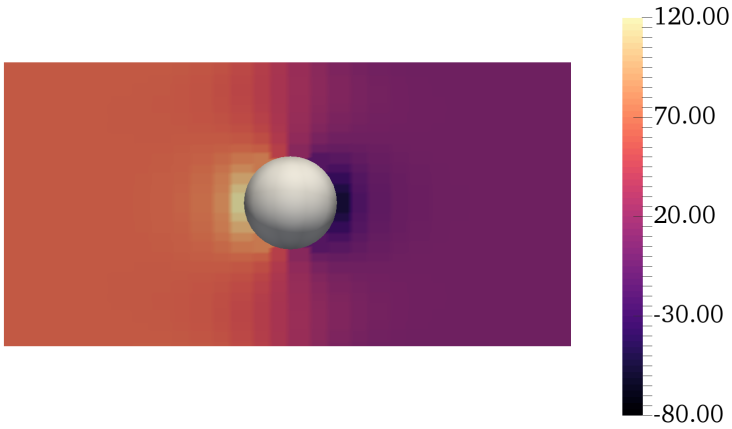
Figure 6.13: Uniform flow around a sphere.



(a) Slice of FE pressure field (g/cm·s^2, = 0.1 Pa) at $z = 0$.



(b) Slice of IE pressure field (g/cm·s^2, = 0.1 Pa) at $z = 0$.



(c) Slice of total pressure field (g/cm·s^2, = 0.1 Pa) at $z = 0$.

Figure 6.14: Pressure associated with uniform flow solution in Figure 6.13.

Chapter 7

Conclusions and future work

7.1 FE-P^{3}M

The finite-element-based particle-particle–particle-mesh (FE-P^{3}M) method was introduced in Chapter 3 and shown to be effective at constructing electrostatic potentials for the N-body problem. For dynamic application, the conservation properties of the overall scheme are important, such as conservation of energy in molecular dynamics simulations. In the preceding, we did not consider momentum or energy conservation in detail. Indeed, for the formulation as presented, the operators do not satisfy the symmetry discussed by Hockney and Eastwood [5], so exact momentum conservation is not anticipated. Moreover, as the basis functions are not differentiable at the collocation points, straightforward analytical differentiation of the potentials is not always possible.

The nature of the mesh solve in the presented method lends itself to varied boundary conditions since the fundamental formulation of the algorithm does not change when the boundary conditions are changed. As with PME, periodic boundary conditions are the simplest to implement in our method, and require no extra effort beyond creating a finite element matrix that honors the periodic structure of the mesh. As presented in Section 3.1 and demonstrated in Section 3.3, even non-trivial Dirichlet (or Neumann) boundary conditions simply require calculations to allow for any short-range effects already present on the surface before applying the conditions to the finite element problem. Furthermore, there is no restriction to only considering boundary conditions that can be implemented with the method of images.

The method is also extensible without fundamental changes to the non-uniform meshes that are commonly used in finite element discretizations. The main difference for general meshes is in the cost of the method. In this case, the screens still solve (3.10). However, the discussed simplifications of the screen coefficient calculations depend on a regular, rectangular mesh and are not applicable to an unstructured mesh with general quadrilateral elements. Thus, the flexibility of a complex mesh is balanced with the benefits of localizing the mesh cells. In addition, the short-range potential becomes more difficult to generalize due to the myriad of shapes a screen can take based on the elements composing it. Computing accurate

values for the short-range potentials may require quadrature-based evaluations for each pair of interacting charges. However, the locality and structured character of these operations is expected to coincide with high-throughput accelerators.

Another variation of the method is to limit the screens to have support in only the element containing the charge, so that each screen includes only degrees of freedom interior to the element. In addition, screens could be defined through their actions, avoiding screen construction altogether, since only integration against the finite element basis functions is needed.

7.2 FE-IE coupling

High-order finite element–integral equation (FE-IE) coupling for embedded domain problems on non-conforming volume meshes was introduced in Chapter 4. The method was analyzed and demonstrated for the interior and exterior embedded domain problems. An immediate extension of this work is to formulate the method for combinations of Neumann and Dirichlet boundary conditions, both on the embedded domain and on $\partial\hat{\Omega}$.

It was also shown how to couple these two types of embedded domain problems to solve interface problems in Chapter 5. The interface method handles homogeneous and non-homogeneous jump conditions and requires no modification to the FE basis functions. It also achieves high-order convergence (even near the embedded interface) without requiring additional interface conditions. Future work includes further analysis of the $\kappa = -c$ case, a scenario not permitted by the method detailed above. In addition, it would be illustrative to rigorously define the influence of the $\partial\mathcal{R}$ operator in the system and other ways to compute the discrete operator ∂R. For example, the gradient could be projected into the space of the basis functions to ensure continuity.

Finally, the exterior method was extended to the incompressible Stokes equations in Chapter 6. Future work includes investigating time-dependent flows, where the rigid body is moving with time. It is expected that the method is extensible to the "embedded Neumann" case — that is, where the traction on the surface of the rigid body is prescribed instead of the velocity. Another possibility is to extend the interface method of Chapter 5 to the Stokes equations, with fluid, potentially of differing viscosity, on either side of an elastic membrane. From a performance standpoint, there are also many opportunities for parallelism in the code in order to perform larger numerical tests.

References

[1] R. Kress, *Linear Integral Equations* (Springer Science+Business Media, New York), second edition (1999).

[2] K. Atkinson and W. Han, *Theoretical Numerical Analysis: A Functional Analysis Framework* (Springer Science+Business Media, New York), third edition (2009).

[3] P. G. Ciarlet, "Basic error estimates for elliptic problems," Handbook of Numerical Analysis, Vol.2, Finite Element Methods (Part 1) pages 17–351 (1991).

[4] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," J. Comp. Phys. **135**, 280 (1987).

[5] R. W. Hockney and J. W. Eastwood, *Computer simulation using particles* (Institute of physics publishing, Bristol) (1988).

[6] T. Darden, D. York and L. Pedersen, "Particle mesh Ewald: An $n\log(n)$ method for Ewald sums in large systems," J. Chem. Phys. **98**, 10089 (1993).

[7] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden and H. L. et al., "A smooth particle mesh Ewald method," J. Chem. Phys. **103**, 8577 (1995).

[8] H. Cheng, L. Greengard and V. Rokhlin, "A fast adaptive multipole algorithm in three dimensions," J. Comp. Phys. **55**, 468 (1999).

[9] A.-K. Tornberg and L. Greengard, "A fast multipole method for the three-dimensional Stokes equations," J. Comp. Phys. **227**, 1613 (2008).

[10] H. Wang, T. Lei, J. Li, J. Huang and Z. Yao, "A parallel fast multipole accelerated integral equation scheme for 3D Stokes equations," Int. J. Numer. Meth. Engng **70**, 812 (2007).

[11] S. Veerapaneni, A. Rahimian, G. Biros and D. Zorin, "A fast algorithm for simulating vesicle flows in three dimensions," J. Comp. Phys. **230**, 5610 (2011).

[12] N. N. Beams, L. N. Olson and J. B. Freund, "A finite element based P^3M method for N-body problems," SIAM J. Sci. Comput. **38**, A1538 (2016).

[13] J. Parvizian, A. Düster and E. Rank, "Finite cell method," Comput. Mech. **41**, 121 (2007).

[14] S. Kollmannsberger, A. Özcan, J. Baiges, M. Ruess, E. Rank and A. Reali, "Parameter-free, weak imposition of Dirichlet boundary conditions and coupling of trimmed and non-conforming patches," Int. J. Numer. Meth. Engng **101**, 670 (2015).

[15] R. Glowinski, T.-W. Pan and J. Periaux, "A fictitious domain method for Dirichlet problem and applications," Comput. Method Appl. M. **111**, 283 (1994).

[16] L. Parussini and V. Pediroda, "Fictitious Domain approach with *hp*-finite element approximation for incompressible fluid flow," J. Comp. Phys. **228**, 3891 (2009).

[17] G. Biros, L. Ying and D. Zorin, "A fast solver for the Stokes equations with distributed forces in complex geometries," J. Comp. Phys. **193**, 317 (2003).

[18] A. Mayo, "The fast solution of Poisson's and the biharmonic equations on irregular regions," SIAM J. Numer. Anal. **21** (1984).

[19] T. Rüberg and F. Cirak, "An immersed finite element method with integral equation correction," Int. J. Numer. Meth. Engng (2010).

[20] R. J. LeVeque and Z. Li, "The immersed interface method for elliptic equations with discontinuous coefficients and singular sources," SIAM J. Numer. Anal. **31**, 1019 (1994).

[21] R. Mittal and G. Iaccarino, "Immersed boundary methods," Annu. Rev. Fluid Mech. **37**, 239 (2005).

[22] Z. Li, "The immersed interface method using a finite element formulation," Appl. Numer. Math. **27**, 253 (1998).

[23] Z. Li, T. Lin and X. Wu, "New Cartesian grid methods for interface problems using the finite element formulation," Numer. Math. **96**, 61 (2003).

[24] Y. Gong, B. Li and Z. Li, "Immersed-interface finite-element methods for elliptic interface problems with nonhomogeneous jump conditions," SIAM J. Numer. Anal. **46**, 472 (2008).

[25] X. He, T. Lin and Y. Lin, "Immersed finite element methods for elliptic interface problems with non-homogeneous interface problems," Int. J. Numer. Anal. Mod. **8**, 284 (2011).

[26] S. Adjerid and T. Lin, "A p-th degree immersed finite element for boundary value problems with discontinuous coefficients," Appl. Numer. Math. **59**, 1303 (2009).

[27] R. J. LeVeque and Z. Li, "Immersed interface methods for Stokes flow with elastic boundaries or surface tension," SIAM J. Sci. Comput. **18**, 709 (1997).

[28] Z. Tan, K. M. Lim and B. C. Khoo, "An immersed interface method for Stokes flows with fixed/moving interfaces and rigid boundaries," J. Comp. Phys. **228**, 6855 (2009).

[29] G. J. Wagner, N. Moës, W. K. Liu and T. Belytschko, "The extended finite element method for rigid particles in stokes flow," Int. J. Numer. Meth. Engng **51**, 293 (2001).

[30] C. S. Peskin, "Flow patterns around heart valves: A numerical method," J. Comp. Phys. **10**, 252 (1972).

[31] L. Zhang, A. Gerstenberger, X. Wang and W. K. Liu, "Immersed finite element method," Comput. Methods Appl. Mech. Engrg **193**, 2051 (2004).

[32] J. P. Hernandez-Ortiz, J. J. de Pablo and M. D. Graham, "Fast computation of many-particle hydrodynamic and electrostatic interactions in a confined geometry," Phys. Rev. Let. **98**, 140602 (2007).

[33] A. Kumar and M. D. Graham, "Accelerated boundary integral method for multiphase flow in non-periodic geometries," J. Comp. Phys. **231**, 6682 (2012).

[34] C. Sagui and T. Darden, "Multigrid methods for classical molecular dynamics simulations of biomolecules," J. Chem. Phys. **114**, 6578 (2001).

[35] B. Sandak, "Multiscale fast summation of long-range charge and dipolar interactions," J. Comput. Chem. **22**, 717 (2001).

[36] R. D. Skeel, I. Tezcan and D. J. Hardy, "Multiple grid methods for classical molecular dynamics," J Comput Chem. **23**, 673 (2002).

[37] J. A. Izaguirre, S. S. Hampton and T. Matthey, "Parallel multigrid summation for the N–body problem," J. Parallel Distrib. Comput. **65**, 949 (2005).

[38] L. Greengard and V. Rokhlin, "A new version of the fast multipole method for the Laplace equation in three dimensions," Acta Numerica **6**, 229 (1997).

[39] P. Ewald, "Die berechnung optischer und elektrostatischer gitterpotentiale," Ann. Phys. **369**, 253 (1921).

[40] D. Pekurovsky, "P3dfft: A framework for parallel computations of fourier transforms in three dimensions," SIAM J. Sci. Comput **34**, C192C209 (2012).

[41] H. Gahvari and W. Gropp, "An introductory exascale feasibility study for ffts and multigrid," Parallel and Distributed Processing Symposium, International **0**, 1 (2010).

[42] D. S. Cerutti, R. E. Duke, T. A. Darden and T. P. Lybrand, "Staggered mesh Ewald: An extension of the smooth Particle-Mesh Ewald method adding great versatility," J. Chem. Theory Comput. **5**, 2322 (2009).

[43] D. S. Cerutti and D. A. Case, "Multi-level Ewald: A hybrid multigrid/fast Fourier transform approach to the electrostatic particle-mesh problem," J. Chem. Theory Comput. **6**, 443 (2010).

[44] E. Pollock and J. Glosli, "Comments on p^3m, fmm, and the ewald method for large periodic coulombic systems," Comput Phys Commun **95**, 93 (1996).

[45] J. B. Freund, "Electro-osmosis in a nanometer-scale channel studied by atomistic simulation," J Chem Phys **116**, 2194 (2002).

[46] L. N. Olson, J. B. Schroder and R. S. Tuminaro, "A general interpolation strategy for algebraic multigrid using energy minimization," SIAM Journal on Scientific Computing **33**, 966 (2011).

[47] L. N. Olson, J. Schroder and R. S. Tuminaro, "A new perspective on strength measures in algebraic multigrid," Numerical Linear Algebra with Applications **17**, 713 (2010).

[48] J. J. Heys, T. A. Manteuffel, S. F. McCormick and L. N. Olson, "Algebraic multigrid for higher-order finite elements," J. Comput. Phys. **204**, 520 (2005).

[49] L. Olson, "Algebraic multigrid preconditioning of high-order spectral elements for elliptic problems on a simplicial mesh," SIAM J. Sci. Comput. **29**, 2189 (2007).

[50] A. H. Baker, R. D. Falgout, T. V. Kolev and U. Meier Yang, "Scaling hypre's multigrid solvers to 100,000 cores," in M. W. Berry, K. A. Gallivan, E. Gallopoulos, A. Grama, B. Philippe, Y. Saad and F. Saied, editors, *High-Performance Scientific Computing*, pages 261–279 (Springer London) (2012).

[51] V. E. Henson and U. M. Yang, "Boomeramg: a parallel algebraic multigrid solver and preconditioner," Applied Numerical Mathematics **41**, 155 (2000).

[52] R. Kirby, "Fast simplicial finite element algorithms using Bernstein polynomials," Numerische Mathematik **117**, 631 (2011).

[53] J. W. Eastwood, "Particle simulation methods in plasma physics," Comput Phys Commun **43**, 89 (1986).

[54] A. C. J. Paes, N. M. Abe, V. A. Serrão and A. Passaro, "Simulations of plasmas with electrostatic PIC models using the finite element method," Braz J Phys **33**, 411 (2003).

[55] I. Lashuk, A. Chandramowlishwaran, H. Langston, T.-A. Nguyen, R. Sampath, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin and G. Biros, "A massively parallel adaptive fast multipole method on heterogeneous architectures," Commun ACM **55**, 101 (2012).

[56] A. Klöckner, A. Barnett, L. Greengard and M. O'Neil, "Quadrature by expansion: A new method for the evaluation of layer potentials," J. Comp. Phys. **252**, 332 (2013).

[57] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin and D. Wells, "The `deal.II` Library, Version 8.4," J. Numer. Math. **24** (2016).

[58] W. Bangerth, R. Hartmann and G. Kanschat, "deal.II – a general purpose object oriented finite element library," ACM Trans. Math. Softw. **33**, 24/1 (2007).

[59] A. Klöckner, "pytential," https://github.com/inducer/pytential.

[60] C. L. Epstein, L. Greengard and A. Klöckner, "On the convergence of local expansions of layer potentials," SIAM J. Numer. Anal. **51**, 2660 (2013).

[61] R. Scott, "Optimal L^∞ estimates for the finite element method on irregular meshes," Math. Comput. **30**, 681 (1976).

[62] A. H. Schatz, "A weak discrete maximum principle and stability of the finite element method in L_∞ on plane polygonal domains," Math. Comput. **34**, 77 (1980).

[63] W. N. Bell, L. N. Olson and J. B. Schroder, "PyAMG: Algebraic multigrid solvers in Python v3.0," (2015), Release 3.2.

[64] H. Power and H. Wrobel, *Boundary Integral Methods in Fluid Mechancs* (Computational Mechanics Publications, Southampton, UK) (1995).

[65] C. Pozrikidis, *Boundary integral and singularity methods for linearized viscous flow* (Cambridge University Press, New York) (1992).

[66] F.-K. Hebeker, "Efficient boundary element methods for three-dimensional exterior viscous flows," Numer. Meth. Part. D. E. **2**, 273 (1986).

[67] G. C. Hsiao and R. Kress, "On an integral equation for the two-dimensional exterior Stokes problem," Appl. Numer. Math. pages 77–93 (1985).

[68] A. M. J. Davis and M. E. O'Neill, "Separation in a slow linear shear flow past a cylinder and a plane," J. Fluid Mech. **81**, 551 (1977).

[69] S. Taneda, "Visualization of separating Stokes flows," J. Phys. Soc. Jpn. **46**, 1935 (1979).

[70] Glycerine Producers' Association, "Physical properties of glycerine and its solutions," retrieved from http://www.aciscience.org/docs/ (1963).