High-order numerical methods for scattering problems



Abinand Gopal

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2021

Acknowledgements

First and foremost, I would like to thank Nick Trefethen and Gunnar Martinsson for their outstanding mentorship over the past four years. I have learned innumerable lessons on mathematics and research from both of them, and I can feel their influence on how I think about all aspects of these subjects.

I would also like to thank the entire Numerical Analysis Group at Oxford for much support and many stimulating discussions. I give special thanks to Yuji Nakatsukasa for many enlightening conversations and Cora Cartis for her support in her capacity as my college advisor. I also give a special thanks to my officemates Bogdan Toader, Thomas Roy, Julien Vaes, and Šimon Váry, as well as Matteo Croci and Florian Wechsung, for their friendship and for teaching me much mathematics.

I am grateful to the Clarendon Fund and Alfred Douglas Stone Fund for funding my doctorate and enabling me to come to Oxford in the first place; I feel incredibly fortunate to have received their support.

I am indebted to Endre Süli and Mike O'Neil for serving as my viva examiners and providing me with much useful feedback, which has greatly improved this thesis.

Of course, none of this would be possible without my professors at Virginia Tech, especially Mark Embree and Lizette Zietsman, who sparked my interest in numerical analysis in the first place and continue to serve as sources of inspiration.

Finally, I would like to thank all my family and friends for their abundance of support and encouragement throughout this entire process.

Abstract

Scattering problems are ubiquitous in scientific and engineering applications. These problems can often be straightforwardly modeled using relatively simple linear partial differential equations, such as the Helmholtz equation. The numerical solution of these equations, however, can be challenging due to several features of the problem, such as ill-conditioning, unbounded domains, corner singularities, and oscillatory solutions. In this thesis we present novel numerical methods for the efficient handling of some of these challenges.

We first consider the case of homogeneous scattering on domains with corners and introduce a new solver in this regime, based on results in the rational approximation theory literature. The solver uses an exponentially clustered sum of dipoles to resolve the singularities in the solution and a simple expansion to resolve the smooth part. We show that the convergence is root-exponential with respect to the number of degrees of freedom and illustrate the behavior of the solver in several numerical experiments.

We next turn our attention to the case of inhomogeneous scattering and present a new solver for the Lippmann–Schwinger equation, an integral equation reformulation of the problem. The new solver is based on the hierarchically block separable matrix format and exploits the geometry of the discretization and problem for accelerated inversion. The solver is shown to be effective both as a direct solver and preconditioner in a number of numerical experiments.

For our last contribution we consider broadband applications in homogeneous scattering. We present a new technique for accelerating existing direct solvers in this regime. The technique works by computing basis matrices for all of the low-rank blocks in a rank-structured matrix format and leverages the observation that the resulting ranks are often comparable to those at just the highest frequency. Again, the viability of the method is illustrated in several numerical experiments.

Contents

1 Introduction **1**

2 Solver for homogeneous scattering on domains with corners **4**

2.1 Introduction 4

2.2 Preliminaries 7

2.2.1 Corner singularities 7

2.2.2 Approximation of $|x|$ on $[-1, 1]$ 8

2.2.3 Least-squares problems 9

2.2.4 Hermite integral formula 10

2.3 Laplace solver 10

2.3.1 Algorithm 10

2.3.2 Numerical experiments 15

2.3.3 Theory 18

2.4 Helmholtz solver 23

2.4.1 Algorithm 23

2.4.2 Numerical experiments 27

2.4.3 Variants 30

2.5 Vekua theory 36

3 Direct solver for scattering in variable media **39**

3.1 Introduction 39

3.2 Preliminaries 44

3.2.1 Interpolative decomposition 44

3.2.2 High-order discretization 46

3.2.3 Iterative solution 47

3.2.4 Hierarchically block separable matrices 49

3.3 HBS compression 53

3.4 Direct solver 58

3.4.1 Description of direct solver . . . 58
3.4.2 Complexity estimate . . . 63
3.4.3 Comparison to other direct solvers . . . 67
3.5 Using the solver as a preconditioner . . . 68
3.6 Numerical results . . . 69
3.6.1 Direct solution for a Gaussian bump . . . 70
3.6.2 Direct solution for a cavity . . . 72
3.6.3 Preconditioned iterative solution for a cavity . . . 74
3.6.4 Preconditioned iterative solution for other geometries . . . 76

4 Broadband recursive skeletonization 80
4.1 Introduction . . . 80
4.2 Preliminaries . . . 83
4.2.1 Discretization . . . 83
4.2.2 Recursive skeletonization . . . 85
4.2.3 Prolate spheroidal wave functions . . . 86
4.3 Theoretical foundations . . . 90
4.3.1 Rank behavior at a single frequency . . . 90
4.3.2 Introduction to universal skeletons . . . 92
4.3.3 Universal skeletons . . . 95
4.3.4 Bandlimited fields . . . 102
4.4 Efficient algorithms for building universal skeletons . . . 107
4.5 Applications and numerical experiments . . . 111
4.5.1 2D boundary integral equation . . . 112
4.5.2 2D eigenvalue problems . . . 119
4.5.3 2D fast matrix-vector products . . . 121
4.5.4 3D boundary integral equation . . . 123

5 Conclusions 126

Bibliography 128

Chapter 1

Introduction

The central focus of this thesis is the numerical solution of Helmholtz equations of the form

$$-\Delta u(\boldsymbol{x}) - \kappa^2(1 - b(\boldsymbol{x}))u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega \subseteq \mathbb{R}^d, \tag{1.1}$$

which frequently arise in the study of time-harmonic wave propagation. While many of the ideas that we discuss are immediately applicable to other areas such as electromagnetics and quantum mechanics, here we limit ourselves to acoustics. In this case u is the total pressure field, $\kappa > 0$ is the free-space wavenumber, and b is a scattering potential, which encodes the deviation in the speed of sound in a medium from that of free space, and relevant applications include seismology, ultrasound imaging, and sonar [44].

Given the scope and importance of these applications, faster and more robust numerical methods are of continual interest. While (1.1) is a relatively simple linear, elliptic partial differential equation, it can possess several features, often stemming from the indefinite nature of the differential operator, that are challenging for numerical methods. For example, equations of the form (1.1) can be intrinsically ill-conditioned, especially when κ is near a resonance. This can necessitate computing high accuracy solutions and can complicate the convergence of iterative solvers [58]. Solutions to these equations also tend to grow increasingly oscillatory as the wavenumber increases, and, as a result, many degrees of freedom may be needed to discretize high-frequency problems. This is exacerbated in finite elements and finite differences by the need to control dispersion errors, which are especially pronounced in low-order methods [12]. Also, we are frequently interested in applications where the domain is unbounded and a radiation condition, such as the Sommerfeld condition, is imposed at infinity. When finite differences, finite elements, or spectral methods are used, this can require special truncated boundary techniques, such as absorbing boundary conditions [54] or

perfectly matched layers [18], which can be challenging to implement and introduce non-physical behavior. Finally, in many environments we are interested in domains with corners, which often lead to singularities in the solutions [77, 109]. These in turn can reduce the convergence rate of numerical methods.

Our contention is that these issues can often be elegantly handled by high-order, global methods. By global methods, we specifically mean approaches based on integral equation reformulations and the closely related method of fundamental solutions. These methods possess a host of nice properties that make them particularly well-suited for these applications. For example, it is often easy to get high-order convergence, which can allow for a reduction in the number of degrees of freedom required for discretization [84]. Moreover, when integral equation methods are used, this can be done efficiently without introducing spurious ill-conditioning [29]. By including the free-space fundamental solution of the Helmholtz equation in the ansatz, these methods automatically satisfy the Sommerfeld condition and prevent dispersion errors [44]. An important trade-off is that since these methods are of a global nature, they result in dense coefficient matrices. Indeed, the solution of dense least-squares and linear systems using direct methods is another recurring theme in this work.

Of course, the use of integral equation methods and the method of fundamental solutions for solving elliptic partial differential equations and scattering problems, in particular, has a long, rich history (see [43, 44, 59] for some discussions on the history of these methods). Here we build upon this existing work in three specific cases: two-dimensional problems on domains with corners, inhomogeneous scattering in the plane, and broadband applications.

In Chapter 2 we consider the case of low-frequency, homogeneous (so that $b = 0$ in (1.1)) scattering on domains with corners in two dimensions. Our approach is inspired by a result of D. J. Newman from 1964 in the rational approximation theory literature and is closely related to the method of fundamental solutions. It uses an ansatz consisting of sums of exponentially clustered dipoles to resolve the singularities and a series expansion to resolve the smooth part of the solution. The coefficients are then determined using linear least squares to fit the boundary data, and accuracy is guaranteed in the interior, all the way up to the boundary, by the maximum principle. The resulting method converges root-exponentially with respect to the number of degrees of freedom, a fact that we have rigorous theory for in the Laplace case and empirical evidence for in the Helmholtz case. The solver is exceptionally easy to implement and can be easily extended to a variety of regimes. We illustrate this latter point with a number of examples.

Our focus shifts to inhomogeneous scattering in the plane in Chapter 3. We now reformulate the scattering problem as an integral equation known as the Lippmann–Schwinger equation. We then discretize the Lippmann–Schwinger equation on uniform grids using locally-corrected trapezoidal rules known as Duan-Rokhlin quadratures. We develop a new solver for inverting the resulting coefficient matrices, based on recent work in the fast direct solvers literature. The solver exploits the uniform grids to significantly accelerate the compression of the coefficient matrix into the hierarchically block separable matrix format. Inversion is then done using a scattering matrix formulation, which is empirically and heuristically demonstrated to be more stable than some previous solvers based on the Woodbury formula. The solver is found to be especially effective as a preconditioner for an iterative solver. All of this is demonstrated by a few numerical experiments, including an example where a problem with a highly refractive scattering potential in a computational domain over 500 wavelengths across is solved to a residual error of less than 10^{-10} with 10^{th}-order quadratures in a little over a couple of hours on a workstation.

In Chapter 4 we again return to homogeneous scattering but this time in both two and three dimensions. Our focus here is on broadband applications where solutions to the Helmholtz equation are desired for many different wavenumbers in some interval. We introduce a new method for accelerating direct solvers using the so-called recursive skeletonization technique in this regime. The method operates by computing a "universal basis" for the column or row space of each block that must be compressed in the rank-structured format; the universal bases are valid over an entire interval of wavenumbers, and so a compressed, rank-structured matrix can be obtained at any wavenumber in the interval simply by projecting the necessary blocks onto their respective universal bases, which is significantly faster than computing the low-rank approximations from scratch. This comes at the expense of increasing the ranks in the rank-structured format, but we present empirical evidence and a heuristic argument that suggest this increase in ranks is only slight for a broad class of geometries. We illustrate the efficacy of our method in a few different environments using numerical experiments.

We end on Chapter 5, where we summarize this work and outline some directions for future work, a few of which are currently being explored.

Chapter 2

Solver for homogeneous scattering on domains with corners[1]

2.1 Introduction

A classical problem in numerical analysis concerns the solution of elliptic partial differential equations (PDEs) on domains with corners. It is well-known that in this regime, the solutions often develop singularities at the corners, even when the boundary data are smooth. In this chapter we will specifically be interested in homogeneous Helmholtz equations on such domains in the plane. A prototypical example of this is the case of exterior, sound-soft scattering where we seek a solution $u \in C^2(\mathbb{R}^2 \backslash \overline{\Omega}) \cap C(\mathbb{R}^2 \backslash \Omega)$ satisfying

$$\begin{cases} -\Delta u(\boldsymbol{x}) - \kappa^2 u(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \mathbb{R}^2 \backslash \overline{\Omega}, \\ u(\boldsymbol{x}) = -u_{\text{inc}}(\boldsymbol{x}), & \boldsymbol{x} \in \Gamma, \\ \sqrt{r}|\partial_r u(\boldsymbol{x}) - i\kappa u(\boldsymbol{x})| = 0, & r = \|\boldsymbol{x}\| \to \infty \end{cases} \tag{2.1}$$

where $\kappa \geq 0$ is the wavenumber, $\Omega \subset \mathbb{R}^2$ is a scatterer, Γ is the boundary of Ω, and $u_{\text{inc}} : \Gamma \to \mathbb{C}$ is the restriction of an incident field to Γ. Throughout this chapter, unless otherwise stated, we will assume that Ω is a bounded, simply connected domain and Γ is a closed, piecewise analytic curve with a finite number of corners, all of which are non-degenerate (no slits or cusps). The third condition in (2.1) is the Sommerfeld radiation condition, which specifies that waves propagate outward toward infinity.

[1] Much of the content in this chapter was first presented in the brief report [66] and the publication [68] both by AG and his doctoral supervisor Lloyd N. Trefethen (LNT). The idea of using rational functions to solve Laplace problems was proposed by LNT, and the idea of using rational functions with fixed poles was suggested by Kirill Serkh during a joint discussion with AG and LNT. AG and LNT jointly developed the Laplace solver and corresponding theory in Section 2.3, and AG developed the extension to the Helmholtz equation after an initial proof of concept by LNT. All of the codes used for numerical experiments in this chapter were developed by AG.

Typically, the incident field is smooth, often given by either a point source or a plane wave, but the solutions develop singularities at the corners nonetheless.

As a result, a naïve application of finite differences, finite elements, boundary integral equations, or spectral methods on (2.1) would result in slow convergence, which can be particularly problematic for applications in scattering given the potential intrinsic ill-conditioning of the physics. While the convergence can be improved in most of these methods, the necessary modifications can be challenging to implement in practice. For example, one could switch to an *hp*-adaptive version of finite elements, which would yield a super-algebraically convergent method [133], but this would require volume meshing, adaptive mesh refinement, and, for exterior problems, special truncated boundary techniques (e.g., [18, 54]). Similarly, for boundary integral equations one could use Serkh-Rokhlin quadratures [94, 134, 135], which would require a sophisticated generalized Gaussian quadrature algorithm, or the recursively compressed inverse preconditioning technique [85], which can be challenging to implement. After careful implementation these methods have all been shown to be robust in practice, and it is typically possible to obtain solvers that are of linear complexity with respect to the number of degrees of freedom in the discretization. Another method that has been demonstrated to be viable for these problems is the nonpolynomial finite element introduced in [15]. This method is simpler to implement than the aforementioned approaches but has the disadvantages of requiring analysis of the singularities and also those generally associated with methods that use non-global representations of the solution.

We will instead take an approach closely related to the method of fundamental solutions (MFS), also known as the charge simulation method (see, e.g., [14, 59, 96, 98]). In the MFS a linear combination of fundamental solutions of the PDE, which can be interpreted as a sum of point sources, is used as an ansatz for the solution. The coefficients are usually determined through some type of least-squares fitting with the boundary data. This method has has been frequently employed in the engineering literature [50, 59], and it has been rigorously established that for a smooth, interior Laplace problem on a domain with analytic boundary, the MFS converges exponentially with respect to the number of degrees of freedom when the sources are placed on a level set of the conformal map from the exterior of the domain to the exterior of the unit disk, assuming that there are no singularities in the conformal map between the level set and the boundary of the domain [97]. This is also conjectured to be the case for Helmholtz problems [14]. Recently, it has been empirically observed that the MFS can obtain super-algebraic convergence for problems with corner singularities when the

sources are placed on panels that are geometrically refined at the corners [110, Ch. 5]. This is analogous to treating singularities in collocation-based boundary integral equation methods through brute force, and, as such, the constants in the scaling can be prohibitively large for problems with several corners.

We take a different perspective, motivated by a result in the rational approximation theory literature by D. J. Newman in 1964. The new method uses a global ansatz, where the singularities in the solution are approximated with exponentially clustered dipoles, and the smooth part is approximated with a standard series expansion. In the case of the Laplace equation, this corresponds to representing the solution as the real part of a rational function. A linear least-squares problem is then solved to fit the boundary data. Accuracy in the computed solution is guaranteed over the entire domain by the maximum principle. The method is observed to converge root-exponentially with respect to the number of degrees of freedom, and we have a rigorous theoretical justification for this in the case of the Laplace equation on convex polygons. The solver for the Helmholtz equation is derived through analogy with the Laplace case and is also empirically observed to converge root-exponentially, albeit we are yet to rigorously establish this. While the method uses cubic linear algebra, the constants are favorable, making it competitive with asymptotically optimal solvers for small-to-medium scale problems. Furthermore, the method is robust and exceptionally easy to implement, requiring no analysis of the singularities.

We now outline the remainder of this chapter. In Section 2.2 we discuss the necessary preliminaries for our solvers. Specifically, we review some standard results on corner singularities and rational approximation theory. We also review some basic facts on the numerical solution of least-squares problems and the Hermite integral formula. Section 2.3 describes our solver for the Laplace case, presents some numerical experiments, and discusses our supporting theory. In Section 2.4 we discuss how our Laplace solver generalizes to a solver for Helmholtz problems. We present more numerical experiments and explain how our solver can be adapted to a few other environments. We close this chapter on Section 2.5; here we describe a potential approach to establishing a rigorous theory for the Helmholtz solver, the detailed pursuit of which is deferred to future work.

2.2 Preliminaries

2.2.1 Corner singularities

The analysis of corner singularities that arise in the solution of elliptic equations is well-studied in the PDEs literature. However, most of the results in this area are very theoretical and concern the precise spaces the solutions live in under modest assumptions on the regularity of the boundary data. We refer the reader to the well-known text [77] for a survey of results of this nature. Here we state a classical result, paraphrased from Theorem 3.3 in [109], that states the leading-order singular behavior of solutions to second-order elliptic PDEs on domains with corners for problems with analytic boundary data under some mild assumptions on the growth of the derivatives of the solution.

Theorem 2.1 (Lehman, 1959)**.** *Let $\Omega \subset \mathbb{R}^2$ be a domain, the boundary of which consists of two analytic arcs Γ_1 and Γ_2 in a neighborhood of the origin. Suppose Γ_1 and Γ_2 meet at the origin to form a corner with interior angle $\alpha\pi$ for some $\alpha > 0$, and suppose the origin is a regular point of both curves. Further suppose that the tangent of Γ_1 at the origin goes into the tangent of Γ_2 at the origin by a positive rotation through the angle $\alpha\pi$. Consider the Dirichlet problem where we seek $u \in C^2(\Omega) \cap C(\Omega \cup \Gamma_1 \cup \Gamma_2)$ such that*

$$\begin{cases} \Delta u(\boldsymbol{x}) + a(\boldsymbol{x})[\partial_1 u](\boldsymbol{x}) + b(\boldsymbol{x})[\partial_2 u](\boldsymbol{x}) + c(\boldsymbol{x})u(\boldsymbol{x}) = f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) = \varphi_1(\boldsymbol{x}), & \boldsymbol{x} \in \Gamma_1, \\ u(\boldsymbol{x}) = \varphi_2(\boldsymbol{x}), & \boldsymbol{x} \in \Gamma_2 \end{cases} \tag{2.2}$$

where a, b, c, f, φ_1, and φ_2 are analytic in both variables in a neighborhood of the origin and $\varphi_1(\mathbf{0}) = \varphi_2(\mathbf{0})$. In addition, assume that as $r = \|\boldsymbol{x}\| \to 0$ in $\Omega \cup \Gamma_1 \cup \Gamma_2$ u satisfies

$$\begin{cases} [\partial_1 u](\boldsymbol{x}) = o(r^{-2+\delta}), & \alpha \leq 1, \\ [\partial_1 u](\boldsymbol{x}) = o(r^{-1/\alpha - 1 + \delta}), & \alpha > 1, \end{cases} \text{ and } \begin{cases} [\partial_2 u](\boldsymbol{x}) = o(r^{-2+\delta}), & \alpha \leq 1, \\ [\partial_2 u](\boldsymbol{x}) = o(r^{-1/\alpha - 1 + \delta}), & \alpha > 1 \end{cases} \tag{2.3}$$

for some $\delta > 0$. Then

$$u(\boldsymbol{x}) = u_1(\boldsymbol{x}) + u_2(\boldsymbol{x}) \tag{2.4}$$

where u_1 and its partial derivatives of all orders remain bounded when $\boldsymbol{x}$ approaches the origin, and

$$u_2(\boldsymbol{x}) = \mathcal{O}(r^{1/\alpha} \log r) \tag{2.5}$$

when $\alpha = 1/m$ *for some* $m \in \mathbb{N}$ *and*

$$u_2(\boldsymbol{x}) = \mathcal{O}(r^{1/\alpha}) \tag{2.6}$$

otherwise. These relations may be indefinitely formally differentiated.

2.2.2 Approximation of $|x|$ on $[-1, 1]$

It is well-known that the errors in both polynomial and rational best approximation to an analytic function on an interval decrease exponentially with respect to the degree, and in this regime, there is generally little benefit to working with rational functions over polynomials. Similarly, it is known that there exist adversarial examples in each differentiability class for which best polynomial and rational approximations converge at the same rate [111, Theorem 7.5 in Ch. 7]. However, this is emphatically not the case when one is interested in approximation for classes of functions with certain types of isolated singularities. The prototypical example of this is illustrated in the classical problem of approximation to the function $|x|$ on $[-1, 1]$. In 1914 Bernstein showed that polynomial best approximation for this problem converges merely at a rate reciprocal to the degree [19]. In precise terms, he established that

$$\min_{p \in \mathcal{P}_n} \max_{x \in [-1,1]} ||x| - p(x)| = \mathcal{O}(n^{-1}) \tag{2.7}$$

as $n \to \infty$ where $\mathcal{P}_n$ denotes the space of all polynomials of degrees less than or equal to n. As such, one must take a polynomial of degree in the thousands in order to obtain an approximation to just a few digits of accuracy.

In 1964 D. J. Newman published a remarkable result that showed by switching to approximations with rational functions, one can actually obtain convergence that is *root-exponential* with respect to the degree. The following theorem is the main result in [124].

Theorem 2.2 (Newman, 1964)**.** *There exist degree* n *rational functions* $\{r_n\}_{n=1}^{\infty}$ *(i.e.,* $r_n = p_n/q_n$ *with* $p_n, q_n \in \mathcal{P}_n$*) and a constant* $C > 0$ *such that*

$$\max_{x \in [-1,1]} ||x| - r_n(x)| = \mathcal{O}(e^{-C\sqrt{n}}) \tag{2.8}$$

as $n \to \infty$*.*

While here we have considered approximation to the specific function $|x|$, this result is indicative of the more general advantage of rational functions over polynomials for approximating functions with a wide class of isolated singularities. For example, a

change of variables shows that approximating $|x|$ on $[-1,1]$ is equivalent to approximating $\sqrt{x}$ on $[0,1]$. With this modification, Theorem 2.2 shows that rational functions can efficiently approximate functions with branch point singularities. The essential feature of Newman's rational approximations that enables this fast convergence is that their poles cluster exponentially at the branch point with spacing that scales $\mathcal{O}(n^{-1/2})$ (i.e., poles that cluster like $\mathcal{O}(e^{-\sigma j/\sqrt{n}})$ for $j = 1, \ldots, n$ and some $\sigma > 0$). Similar root-exponential approximation results have also been established for related functions such as x^α on $[0,1]$ [64, 143] and z^α in subsets of $\mathbb{C}$ for $\alpha > 0$ [67, 145]. All of these cases feature rational approximations with poles exponentially clustered at the singularity with $\mathcal{O}(n^{-1/2})$ or closely related spacing. This was studied in-depth, along with the slightly modified spacing we use in our algorithms, in [153]. For a more complete treatment on the history of the problem of approximating $|x|$, we refer the reader to [152, Ch. 25].

Remark 2.1. The optimal constants in (2.8) were conjectured by Varga, Ruttan, and Carpenter in [156] and proven by Stahl in [142]. The resulting rate can be characterized by

$$\lim_{n\to\infty} e^{\pi\sqrt{n}} \left[\min_{\substack{r_n = p_n/q_n \\ p_n, q_n \in \mathcal{P}_n}} \max_{x\in[-1,1]} ||x| - r_n(x)| \right] = 8. \tag{2.9}$$

2.2.3 Least-squares problems

Our solver relies on the solution of a typically extremely ill-conditioned least-squares problem. However, for our purposes we care only about computing a small-residual solution and not the actual forward error between the computed solution and the true minimizer to the least-squares problem. The following lemma states that a computed solution with near-optimal residual can be computed if a backward stable algorithm is used and the norms of both the true and computed solutions are modest in size. It is a simple consequence of the triangle inequality, and statements of it can be found in various places in the literature, for example [15, Lemma 7.1].

Lemma 2.3. *Suppose $\mathbf{x} \in \mathbb{C}^n$ solves the least-squares problem*

$$\min_{\mathbf{x}\in\mathbb{C}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \tag{2.10}$$

for some $\mathbf{A} \in \mathbb{C}^{m\times n}$ and $\mathbf{b} \in \mathbb{C}^m$. Let $\widehat{\mathbf{x}} \in \mathbb{C}^n$ be a computed solution to (2.10) that satisfies

$$\min_{\widehat{\mathbf{x}}\in\mathbb{C}^n} \|(\mathbf{A} + \Delta\mathbf{A})\widehat{\mathbf{x}} - (\mathbf{b} + \Delta\mathbf{b})\| \tag{2.11}$$

for some $\Delta\mathbf{A} \in \mathbb{C}^{m\times n}$ *and* $\Delta\mathbf{b} \in \mathbb{C}^m$*. Then* $\widehat{\mathbf{x}}$ *satisfies*

$$\|\mathbf{A}\widehat{\mathbf{x}} - \mathbf{b}\| \leq \|\mathbf{A}\mathbf{x} - \mathbf{b}\| + (2 + \|\mathbf{x}\| + \|\widehat{\mathbf{x}}\|)\max(\|\Delta\mathbf{A}\|, \|\Delta\mathbf{b}\|). \tag{2.12}$$

2.2.4 Hermite integral formula

A key tool in establishing theory for our Laplace solver is the *Hermite integral formula*, which allows one to express the error between an analytic function and its rational interpolant with prescribed interpolation points and poles explicitly as a contour integral. The following theorem appears as Theorem 2 in [163].

Theorem 2.4 (Walsh, 1932)**.** *Let* $\Omega \subset \mathbb{C}$ *be a simply connected domain bounded by a closed curve* Γ*, and let* f *be analytic in the region enclosed by* Γ *and extend continuously to* Γ*. Also, let distinct points* $\{\alpha_j\}_{j=1}^n \subset \Omega$ *and* $\{\beta_j\}_{j=1}^n \subset \mathbb{C}\backslash\overline{\Omega}$ *be given, and let* r *denote the unique type* $(n-1, n)$ *rational function (i.e.,* $r = p/q$ *for some* $p \in \mathcal{P}_{n-1}$ *and* $q \in \mathcal{P}_n$*) with simple poles at* $\{\beta_j\}_{j=1}^n$ *that interpolates* f *at* $\{\alpha_j\}_{j=1}^n$*. Then the error in the interpolation is given by*

$$f(z) - r(z) = \frac{1}{2\pi i}\int_\Gamma \frac{\phi(z)}{\phi(t)}\frac{f(t)}{t-z}\,dt, \quad z \in \Omega \tag{2.13}$$

where

$$\phi(z) = \prod_{j=1}^n (z-\alpha_j) \Bigg/ \prod_{j=1}^n (z-\beta_j). \tag{2.14}$$

2.3 Laplace solver

2.3.1 Algorithm

We begin by developing our solver for the Laplace equation, which we colloquially refer to as the "lightning Laplace solver". For our purposes it will be convenient to work in $\mathbb{C}$ instead of $\mathbb{R}^2$. To this end, let $\Omega \subset \mathbb{C}$ be a bounded, simply connected domain with boundary Γ, a closed, piecewise analytic curve. We assume that Γ has a finite number of corners, all of which are non-degenerate. Let boundary data $h : \Gamma \to \mathbb{R}$ be given where h is assumed to be real-analytic on segments of Γ in between corners and continuous at the corners. We will initially restrict ourselves to the interior Dirichlet problem. Specifically, we seek $u \in C^2(\Omega) \cap C(\overline{\Omega})$ such that u satisfies

$$\begin{cases} \Delta u(z) = 0, & z \in \Omega, \\ u(z) = h(z), & z \in \Gamma. \end{cases} \tag{2.15}$$

From Theorem 2.1 we expect that in this setting u will typically possess branch point singularities at the corners of Ω. The slow convergence of straightforward discretizations can then be viewed as a matter of approximation theory: standard finite elements, finite differences, and spectral methods are all based on some type of polynomial approximation, and polynomials cannot efficiently approximate functions with such singularities. However, in Section 2.2.2 we found that under similar conditions, rational functions can often achieve much faster convergence. This motivates us to instead use a discretization based on rational functions.

To the best of our knowledge, the only prior work that has pursued a similar direction in this regime is [91]. There, Hochman, Leviatan, and White developed an iterative solver where at each iteration a nonlinear rational fitting algorithm is used to determine pole locations, and then a linear least-squares solver is used to fit the boundary data. This approach was shown to be capable of achieving high accuracy for a variety of problems relatively quickly but suffers some issues common to nonlinear solvers, such as non-monotonic convergence and a general lack of robustness with no theoretical guarantees. For problems where there are singularities near the domain that are unknown a priori, having to determine pole locations through some type of nonlinear fitting is perhaps inevitable. However, under our assumptions, we know that the singularities will be precisely located at the corners of Ω, a fact which we can exploit.

Indeed, it turns out that we can remain in the realm of linear approximation by exploiting the key idea in Theorem 2.2 in Section 2.2.2, namely that rational functions can approximate the behavior of functions with branch point singularities if the poles are exponentially clustered with an appropriate spacing. In particular, to solve a problem of the form (2.15), we represent the solution using an ansatz of the form

$$u(z) \approx \operatorname{Re}\left[\sum_{j=1}^{N_1} \frac{a_j}{z - z_j} + \sum_{j=0}^{N_2} b_j (z - z_*)^j\right] \tag{2.16}$$

where $z_* \in \Omega$ and $\{z_j\}_{j=1}^{N_1} \subset \mathbb{C}\backslash\overline{\Omega}$ are fixed to only depend on the geometry, and $\{a_j\}_{j=1}^{N_1} \subset \mathbb{C}$ and $\{b_j\}_{j=0}^{N_2} \subset \mathbb{C}$ are determined by fitting the boundary data on Γ using linear least squares. In (2.16) the first term is a sum of partial fractions and is included to resolve the singularities in the solution. The second term is a simple polynomial expansion centered at the point z_* and is used to approximate the remaining, smooth part of the solution. We can show that as long as the poles of the approximation $\{z_j\}_{j=1}^{N_1}$ are fixed to cluster exponentially with a square root spacing similar to that in Theorem 2.2, root-exponential convergence with respect to the number of degrees of

freedom can be obtained. Since (2.16) is the real part of an analytic function on Ω, we are guaranteed that the ansatz is harmonic and therefore automatically satisfies the Laplace equation. Thus, if we can fit the boundary data on a suitable discretization of Γ to a certain tolerance, then we are guaranteed accuracy in the interior of Ω to that same tolerance by the maximum principle.

For the purpose of concreteness, suppose Ω has p corners located at $\{\zeta_k\}_{k=1}^p$, and let $\varepsilon > 0$ be the user-specified error tolerance. We begin by fixing a parameter $n \in \mathbb{N}$ which controls the number of degrees of freedom and will be incrementally increased to achieve convergence. For each corner located at ζ_k for $k = 1, 2, \ldots, p$, we determine the locations of $n_k = \mathcal{O}(n)$ poles $\{z_j^{(k)}\}_{j=1}^{n_k}$ in $\mathbb{C}\backslash\overline{\Omega}$ such that they cluster exponentially at ζ_k. In precise terms, for each $k = 1, 2, \ldots, p$, we choose $\{z_j^{(k)}\}_{j=1}^{n_k}$ so that they lie in a straight line and accumulate at ζ_k with spacing

$$|z_j^{(k)} - \zeta_k| = Ce^{-\sigma(\sqrt{n_k}-\sqrt{j})}, \quad j = 1, 2, \ldots, n_k \tag{2.17}$$

where $C > 0$, depending only on Ω, is introduced to make our ansatz scale invariant and $\sigma > 0$ affects the constants in the convergence rate (in our numerics we find that $\sigma = 4$ is a good choice). Note that we could have chosen the same $\mathcal{O}(n^{-1/2})$ spacing as in the rational functions in Theorem 2.2 and still obtain root-exponential convergence, and this is in fact the regime in which we established our theory. However, we empirically found in [68] that the constants are slightly better with the clustering in (2.17), which still has a similar square root spacing. Later, this choice of spacing was studied in detail and justified in [153].

The angle at which each collection of poles meets its respective corner is unimportant for the convergence rate, and in practice we simply place the poles on the outward bisector at each corner. The precise relation between n_k and n also affects the constants in the convergence order. Roughly speaking, we want n_k to be greater near corners with stronger singularities. While it is possible to adaptively choose n_k, here we simply choose $n_k = n$ when the corner at ζ_k is salient and $n_k = 3n$ when it is reentrant. We let $N_1 = \sum_{k=1}^p n_k$ and $\{z_j\}_{j=1}^{N_1} = \cup_{k=1}^p \{z_j^{(k)}\}_{j=1}^{n_k}$ (the ordering does not matter). We choose the number of terms in our polynomial series, N_2, so that $N_2 = \mathcal{O}(n)$ (in our codes, we choose $N_2 = n$) and then set $N = 2N_1 + 2N_2 + 1$, which is the total number of real degrees of freedom. The center of the polynomial expansion z_* is chosen purely based on conditioning considerations. In practice we typically find that choosing $z_* = \frac{1}{p}\sum_{k=1}^p \zeta_k$ works well.

We then discretize Γ using $M = \mathcal{O}(N)$ points (in our numerics we find that we can take $M \approx 4N$) to obtain the collection $\{w_i\}_{i=1}^M$. Some caution must be exercised

to ensure that the discretization is sufficiently dense near the corners to accurately reflect the behavior of the solution, given that the poles cluster exponentially there. For this reason we ensure that the points $\{w_i\}_{i=1}^M$ cluster similarly to (2.17). After this, we can construct our least-squares problem. We form the vector $\mathbf{b} \in \mathbb{R}^M$ with components given by

$$\mathbf{b}(i) = h(w_i) \tag{2.18}$$

and the matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ with entries given by

$$\mathbf{A}(i,j) = \begin{cases} \operatorname{Re}\left[\frac{1}{w_i - z_j}\right], & 1 \le j \le N_1, \\ -\operatorname{Im}\left[\frac{1}{w_i - z_{j-N_1}}\right], & N_1 + 1 \le j \le 2N_1, \\ \operatorname{Re}\left[(w_i - z_*)^{j-2N_1-1}\right], & 2N_1 + 1 \le j \le 2N_1 + N_2 + 1, \\ -\operatorname{Im}\left[(w_i - z_*)^{j-2N_1-N_2-1}\right], & 2N_1 + N_2 + 2 \le j \le N \end{cases} \tag{2.19}$$

for $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$. We now would like to compute the solution to the least-squares problem

$$\mathbf{c} = \underset{\mathbf{x} \in \mathbb{R}^N}{\arg\min} \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \tag{2.20}$$

where the components of $\mathbf{c}$ are related to the coefficients in (2.16) by

$$\mathbf{c}(j) = \begin{cases} \operatorname{Re}\left[a_j\right], & 1 \le j \le N_1, \\ \operatorname{Im}\left[a_{j-N_1}\right], & N_1 + 1 \le j \le 2N_1, \\ \operatorname{Re}\left[b_{j-2N_1-1}\right], & 2N_1 + 1 \le j \le 2N_1 + N_2 + 1, \\ \operatorname{Im}\left[b_{j-2N_1-N_2-1}\right], & 2N_1 + N_2 + 2 \le j \le N \end{cases} \tag{2.21}$$

for $j = 1, 2, \ldots, N$. It is important to note that the matrix $\mathbf{A}$ is exponentially ill-conditioned with respect to n due to the presence of the polynomial term in (2.16). Fortunately, we only care about computing a solution to (2.20) that yields a near-optimal residual and not actually about the forward error between the computed coefficients and the true minimizer. In this regime the relevant result is Lemma 2.3, which states that the suboptimality in the residual with the computed solution is within a factor effectively bounded by the coefficient norms of the true and computed solutions if a backward stable algorithm is used. We find that for the purpose of controlling the relevant coefficient norms, it is preferable to scale $\mathbf{A}$ so that each of its columns has norm 1. To this end, we form the diagonal matrix

$$\mathbf{D} = \operatorname{diag}\left(\|\mathbf{A}(:,1)\|, \|\mathbf{A}(:,2)\|, \cdots \|\mathbf{A}(:,N)\|\right), \tag{2.22}$$

compute the solution to the scaled least-squares problem

$$\widetilde{\mathbf{c}} = \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min} \|\mathbf{A}\mathbf{D}^{-1}\mathbf{x} - \mathbf{b}\|, \tag{2.23}$$

and then compute $\mathbf{c} = \mathbf{D}\widetilde{\mathbf{c}}$. We solve (2.23) using QR factorization via MATLAB's `mldivide` function (\), but any backward stable algorithm would work. After the coefficients have been determined, the error in the computed solution can be estimated by checking $\|\mathbf{A}\mathbf{c} - \mathbf{b}\|_\infty$. By the maximum principle for harmonic functions, this bounds the error in the interior as well. If the convergence criterion $\|\mathbf{A}\mathbf{c} - \mathbf{b}\|_\infty < \varepsilon$ is not met, n is increased, and the process is repeated. This procedure is summarized in Algorithm 2.1.

The solution obtained through this process can then be evaluated at a set of points $\{t_i\}_{i=1}^P \subset \mathbb{C}$ in $\mathcal{O}(PN)$ complexity by forming the matrix $\mathbf{A}_{t_i,z_j,z_*} \in \mathbb{R}^{P\times N}$ with entries given by the right-hand side of (2.19) with the t_i replacing the w_i for $i = 1, 2, \ldots, P$ and then simply evaluating $\mathbf{A}_{t_i,z_j,z_*}\mathbf{c}$. This also allows us to efficiently compute an a posteriori error estimate by checking the residual on a finer discretization of Γ as well as harmonically continue the solution outside of Ω.

An advantage to working with our solver is that since the solution is given by the simple global representation (2.16), it can easily be modified to handle a broad range of other problems. For example, the exterior Dirichlet problem

$$\begin{cases} \Delta u(z) = 0, & z \in \mathbb{C}\backslash\overline{\Omega}, \\ u(z) = h(z), & z \in \Gamma, \\ u(z) = \mathcal{O}(1), & z \to \infty \end{cases} \tag{2.24}$$

can be handled by simply modifying (2.16) to

$$u(z) \approx \operatorname{Re}\left[\sum_{j=1}^{N_1} \frac{a_j}{z - z_j} + \sum_{j=0}^{N_2} \frac{b_j}{(z - z_*)^j}\right] \tag{2.25}$$

where now the $\{z_j\}_{j=1}^{N_1}$ are in Ω and making the appropriate adjustments to (2.19).

Remark 2.2. The cost of our solver is clearly dominated by solving the least-squares problem (2.23) in the final iteration. When QR factorization is used, the associated cost is $\mathcal{O}(MN^2)$. For a fixed convergence tolerance $\varepsilon > 0$, the root-exponential convergence of our method suggests that we will need to take $N = \mathcal{O}(p|\log\varepsilon|^2)$. Since $M = \mathcal{O}(N)$, it then follows that the complexity of the method, in terms of the number of corners and tolerance, is $\mathcal{O}(p^3|\log\varepsilon|^6)$. Fortunately, the constants in the scaling are extremely favorable as will be evidenced in Section 2.3.2. It is perhaps possible that

the complexity can be accelerated to $\mathcal{O}(N)$ or, equivalently, $\mathcal{O}(p|\log\varepsilon|^2)$ by exploiting rank-structure in the coefficient matrix $\mathbf{A}$ and then using a fast direct solver, such as the one presented in [90] (see [110, Section 6.1]). The challenge here is finding a solver that is sufficiently stable such that a result similar to Lemma 2.3 holds. This is the topic of ongoing work.

Remark 2.3. Here, we have chosen a rather crude strategy for determining the number of poles at each corner, which only distinguishes between salient and reentrant corners. This has the advantages of being simple to explain and implement and was the method used in [68]. Better performance can be achieved using a more sophisticated type of adaptivity where the error near each corner is tracked and determines whether more poles are needed to resolve the singularity. Currently, such a solver called `laplace.m` for interior Laplace problems can be found at [151].

Algorithm 2.1 Solver for planar interior Dirichlet Laplace problems

Input: Boundary Γ with corners at $\{\zeta_k\}_{k=1}^{p}$, boundary data h, center of polynomial expansion z_*, and convergence tolerance $\varepsilon > 0$

Output: Poles $\{z_j\}_{j=1}^{N_1}$ and coefficients $\{a_j\}_{j=1}^{N_1}$ and $\{b_j\}_{j=0}^{N_2}$ in (2.16)

for all n increasing quadratically **do**
  Generate poles $\{z_j\}_{j=1}^{N_1} \subset \mathbb{C}\backslash\overline{\Omega}$ where the poles cluster at $\{\zeta_k\}_{k=1}^{p}$ with spacing (2.17) and $N_1 = \mathcal{O}(pn)$
  Fix $N_2 = \mathcal{O}(n)$, and set $N = 2N_1 + 2N_2 + 1$
  Generate a sample set $\{w_i\}_{i=1}^{M} \subset \Gamma$ that also exponentially clusters at $\{\zeta_k\}_{k=1}^{p}$ where $M = \mathcal{O}(N)$
  Assemble $\mathbf{A} \in \mathbb{R}^{M\times N}$ according to (2.19) and $\mathbf{b} \in \mathbb{R}^{M}$ according to (2.18)
  Form $\mathbf{D}$ in (2.22)
  Compute $\widetilde{\mathbf{c}} = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{D}^{-1}\mathbf{x} - \mathbf{b}\|$ using a backward stable algorithm
  Evaluate $\mathbf{c} = \mathbf{D}\widetilde{\mathbf{c}}$
  if $\|\mathbf{A}\mathbf{c} - \mathbf{b}\|_\infty < \varepsilon$ **then**
    Obtain $\{a_j\}_{j=1}^{N_1}$ and $\{b_j\}_{j=0}^{N_2}$ from $\mathbf{c}$ using (2.21)
    return $\{z_j\}_{j=1}^{N_1}$, $\{a_j\}_{j=1}^{N_1}$, and $\{b_j\}_{j=0}^{N_2}$
  end if
end for

2.3.2 Numerical experiments

We demonstrate our solver on four Laplace Dirichlet problems. Our summary of the results for each problem consists of two plots, appearing in a row in Fig. 2.1. Each left plot illustrates convergence and shows how the residual in the least-squares problem

decreases as the number of degrees of freedom is increased. Note that the vertical axis is logarithmically scaled, and the horizontal axis is scaled by the square root of the number of degrees of freedom. On the right is a plot of the computed solution at the final iteration. The locations of the poles of the final solution $\{z_j\}_{j=1}^{N_1}$ are plotted as red dots, and the center of the polynomial expansion z_* is denoted by a red asterisk. An a posteriori error estimate (E_{fine}), given by the residual evaluated on a mesh four-times as fine as the one used for the final least-squares solution, and the total computation time (t_{total}) are displayed in the bottom left corners of the plots on the left. All timing experiments in this chapter were done using a workstation with 2 Intel Xeon Gold 6254 processors (18 cores at 3.1GHz base frequency) with 768 GB of RAM, and all codes were implemented in MATLAB 2019b. In all cases convergence to the desired tolerance is achieved, and the method is reasonably fast. The convergence curve in every case is almost a straight line, indicating that the method is converging root-exponentially as expected. The four problems considered here are:

L-shaped domain[2]	This is an interior Dirichlet problem on the classical L-shaped domain considered in numerical PDEs, which has 5 salient corners and 1 reentrant corner. The boundary data is given by $h(z) = (\operatorname{Re} z)^2$, and the convergence tolerance is set to 10^{-10}.
Five-pointed star	This is an interior Dirichlet problem on a five-pointed star with 5 salient corners and 5 reentrant corners. The boundary data is given by the point source $h(z) = \log\lvert z\rvert$, and the convergence tolerance is set to 10^{-10}.
United Kingdom[3]	This is an interior Dirichlet problem on a coarse model of the United Kingdom with 40 corners. The boundary data is given by $h(z) = \log\lvert z - 0.1 + 0.4i\rvert$, and the convergence tolerance is set to 10^{-5}.
Plane	This is an exterior Dirichlet problem where Ω is given by a crude 2D model of a plane with 16 corners. The boundary data is given by $h(z) = \log\lvert z + 5 + 5i\rvert$, and the convergence tolerance is set to 10^{-5}.

[2]This problem was posed as a challenge to the NA Digest email list in December 2018 [150].

[3]This geometry was provided by Mikael Slevinsky, who in turn received it from Dave Hewett.

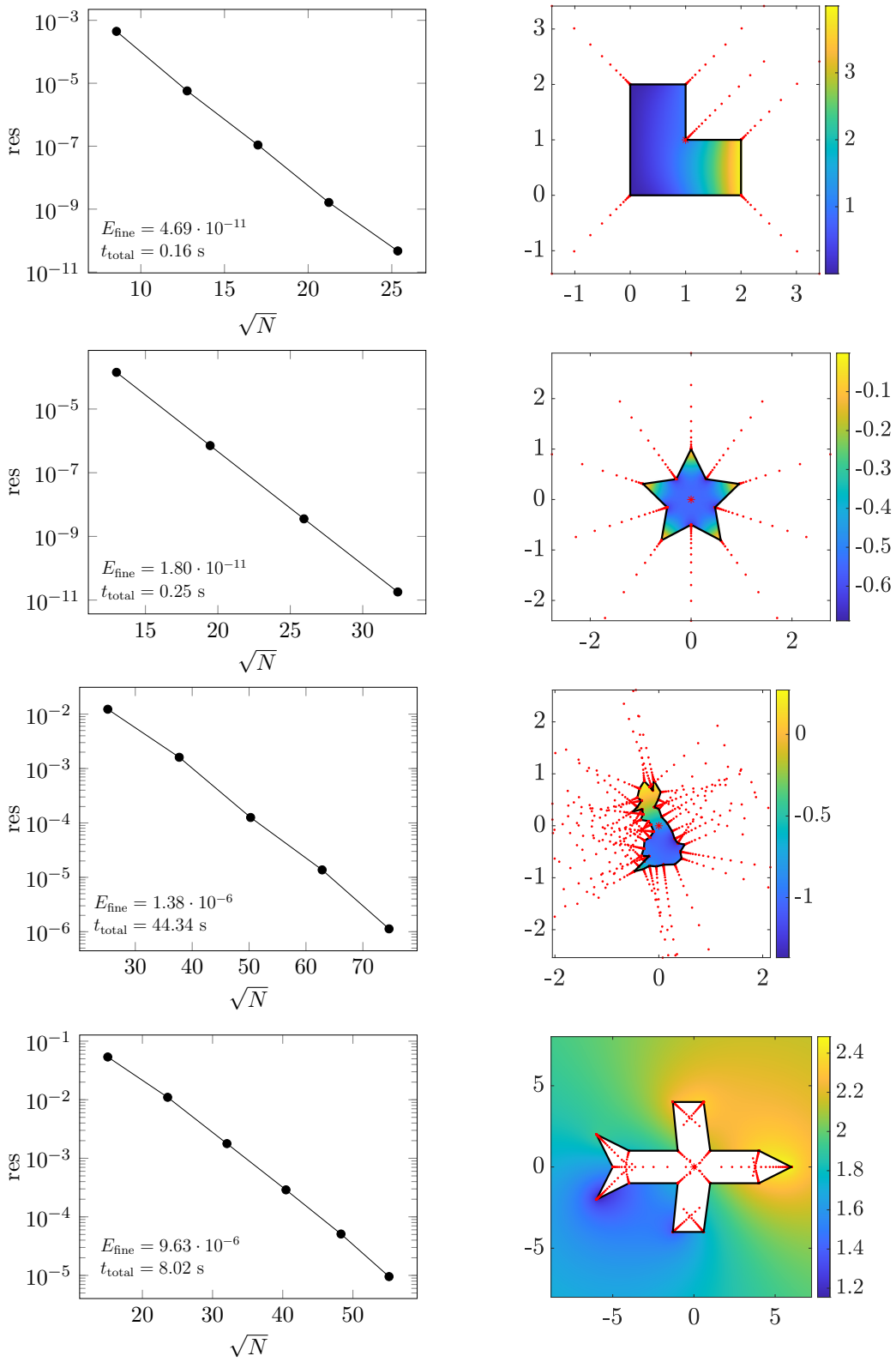


Figure 2.1: Plots for Section 2.3.2

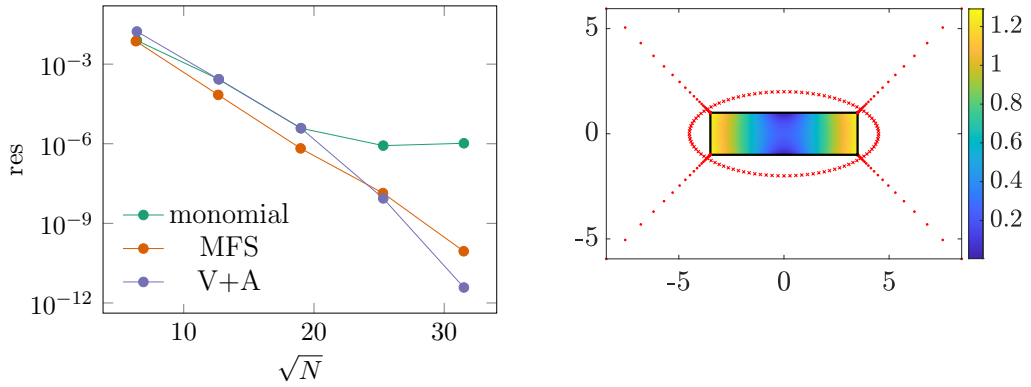


Figure 2.2: (left) Convergence plot for our method when the ansatz (2.16) is used compared to when the monomial part is replaced by an MFS approximation with sources placed on an ellipse and a polynomial generated by the Vandermonde with Arnoldi technique of [30]. (right) Plot of the final solution computed using the MFS-variant. The red ×'s denote the locations of the sources used to resolve the smooth part of the solution.

Remark 2.4. In all of these experiments, we have been able to achieve convergence to the desired tolerance. Generally, we find that the bottleneck for the accuracy achievable is the conditioning of the polynomial term in (2.16). In particular, as n increases the columns of (2.19) associated with the polynomial expansion become increasingly colinear, especially when the monomial basis is poorly scaled for Ω. A robust method for handling this is the Vandermonde with Arnoldi technique introduced in [30]. However, it is currently not clear that this method can be extended to Helmholtz problems. Fortunately, in such cases one can instead usually modify (2.16) to use a representation other than an expansion in the monomial basis to capture the smooth part of the solution. This is illustrated in Fig. 2.2 which shows the results of our solver on an interior Dirichlet problem when $\Omega = (-3.5, 3.5) \times (-1, 1)$ when the MFS with sources placed on an ellipse with major-axis length 4.5 and minor-axis length 1 is used compared with when a polynomial for the smooth part of the solution with and without the Vandermonde with Arnoldi technique. The skewed aspect ratio of the domain causes stagnation in the plain polynomial variant after the residual is about 10^{-6}, whereas the MFS and Vandermonde with Arnoldi variants are capable of achieving a residual below 10^{-10}.

2.3.3 Theory

For the purpose of developing theory, it is typically more convenient to work with analytic functions rather than harmonic functions. Fortunately, we can change an

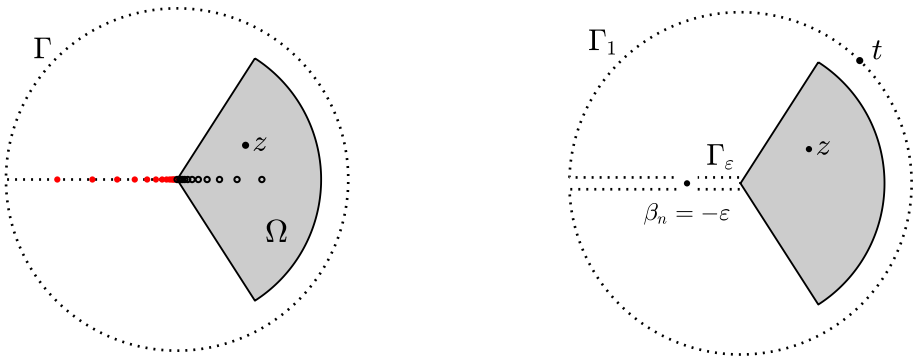


Figure 2.3: Setup for Theorem 2.5 (left) and partitioning of contour Γ in its proof (right)

approximation problem with harmonic functions to one with analytic functions via the harmonic conjugate on simply connected domains. The necessary regularity in the harmonic conjugate can be established through a conformal mapping and appeal to the standard theory of the Hilbert transform (see [87, Theorem 14.2c]). We begin by showing that a function with a single singularity can be locally approximated by rational functions with root-exponential convergence with respect to the degree. We first presented this result as Theorem 2.2 in [68], and a result with a similar flavor can be found in [145, Theorem 2.1a].

Theorem 2.5. *Let f be an analytic function in the slit disk $A = \{z \in \mathbb{C} : |z| < 1\}\backslash(-1, 0]$, satisfying $f(z) = \mathcal{O}(|z|^\delta)$ for some $\delta > 0$ as $z \to 0$, and let $\theta \in (0, \pi/2)$ be given. Then there exist $C > 0$, $\rho \in (0, 1)$, and degree n rational functions $\{r_n\}_{n=1}^\infty$ such that*

$$\max_{z \in \rho\Omega} |f(z) - r_n(z)| = \mathcal{O}(e^{-C\sqrt{n}}) \tag{2.26}$$

as $n \to \infty$ where

$$\Omega = \{z \in \mathbb{C} : |z| < 1 \text{ and } |\arg z| < \theta\}. \tag{2.27}$$

Moreover, each r_n can be chosen to have poles located at

$$\beta_j = -e^{-\sigma(j-1)/\sqrt{n}}, \quad 1 \le j \le n \tag{2.28}$$

for any $\sigma > 0$.

Proof. We will construct the $\{r_n\}_{n=1}^\infty$ to be rational interpolants of f. The relevant error estimate then follows from the Hermite integral formula in Theorem 2.4. To this

end, for each $n > 0$ we introduce interpolation points $\{\alpha_j\}_{j=1}^n$ where

$$\alpha_j = \begin{cases} 0, & j = 1, \\ -\beta_j, & 2 \le j \le n \end{cases} \tag{2.29}$$

for $j = 1, 2, \ldots, n$. We choose r_n to be the unique type $(n-1, n)$ rational interpolant with poles $\{\beta_j\}_{j=1}^n$ given by (2.28) that interpolates f at $\{\alpha_j\}_{j=1}^n$. We take the contour for the Hermite integral formula, Γ, to be the boundary of the slit disk. Without loss of generality, we may assume that f extends continuously to Γ since, if not, we may integrate over contours in A that come arbitrarily close to Γ. This setup is illustrated in the left image of Fig. 2.3. We partition Γ into two disjoint pieces Γ_ε and Γ_1 such that

$$\Gamma_\varepsilon = \{z \in \Gamma : |z| < \varepsilon\} \quad \text{and} \quad \Gamma_1 = \Gamma \backslash \Gamma_\varepsilon \tag{2.30}$$

where $\varepsilon = -\beta_n$ (see right image of Fig. 2.3). We can then subdivide (2.13) into

$$f(z) - r_n(z) = \underbrace{\frac{1}{2\pi i} \int_{\Gamma_\varepsilon} \frac{\phi(z)}{\phi(t)} \frac{f(t)}{t - z}\, dt}_{=I_\varepsilon} + \underbrace{\frac{1}{2\pi i} \int_{\Gamma_1} \frac{\phi(z)}{\phi(t)} \frac{f(t)}{t - z}\, dt}_{=I_1} \tag{2.31}$$

for any $z \in A$. We will bound $|I_\varepsilon|$ and $|I_1|$ separately, starting with the former. First, we note that for z in the right-half plane $|z - \alpha_1|/|z - \beta_1| \le |z|$ and $|z - \alpha_j|/|z - \beta_j| \le 1$ for $2 \le j \le n$, and so

$$|\phi(z)| \le |z|, \quad \operatorname{Re} z \ge 0. \tag{2.32}$$

As such, for any $t \in \Gamma_\varepsilon$ and z with $\operatorname{Re} z \ge 0$, we have

$$\left| \frac{\phi(z)}{t - z} \right| \le 1. \tag{2.33}$$

Now, for $t \in \Gamma_\varepsilon$ we have $|t - \alpha_1|/|t - \beta_1| \ge |t|$ and $|t - \alpha_j|/|t - \beta_j| \ge 1$ for $2 \le j \le n$, and so $|\phi(t)| \ge |t|$. Combining these results gives us the bound

$$|I_\varepsilon| \le \frac{1}{2\pi} \int_{\Gamma_\varepsilon} \left| \frac{\phi(z)}{\phi(t)} \frac{f(t)}{t - z} \right| |dt| \le \int_0^\varepsilon L s^{\delta - 1}\, ds \le \frac{L \varepsilon^\delta}{\delta} \tag{2.34}$$

which holds for any z with $\operatorname{Re} z \ge 0$, for some constant $L > 0$ depending only on f, and sufficiently large n, where we have used the fact that $f(z) = \mathcal{O}(|z|^\delta)$ when $z \to 0$. Since $\varepsilon = -\beta_n = e^{-\sigma(n-1)/\sqrt{n}}$, the right-most quantity in (2.34) is $\mathcal{O}(e^{-\sigma\delta\sqrt{n}})$ as $n \to \infty$.

We must now bound $|I_1|$. Using the boundedness and behavior at the origin of f, we can find a constant $M > 0$ independent of n (and therefore the exact choice of ε) such that

$$\int_{\Gamma_1} \left| \frac{f(t)}{t - z} \right| |dt| < M \tag{2.35}$$

for all $z \in \Omega$. Thus, it suffices to show that $|\phi(z)|/|\phi(t)|$ decays root-exponentially as $n \to \infty$ uniformly for $t \in \Gamma_1$ and $z \in \rho\Omega$ for some choice of $\rho \in (0,1)$. We begin by showing $|\phi(z)|$ is root-exponentially small for all $z \in \Omega$. When $z \in \Omega$ and $|z| \leq 2\varepsilon$, this follows from (2.32). For $z \in \Omega$ with $|z| > 2\varepsilon$, we will show that $\mathcal{O}(\sqrt{n})$ terms in the product in (2.14) can be bounded by a constant $D \in (0,1)$ depending only on the angle θ. This can be done for any $z \in \Omega$ by choosing the terms $|z-\alpha_j|/|z-\beta_j|$ for $|z|/2 < \alpha_j < |z|$. Due to the square-root spacing in (2.28) and (2.29), there will be $\mathcal{O}(\sqrt{n})$ such terms for all $|z| \geq 2\varepsilon$ as $n \to \infty$, and by the geometry of Ω, we can find a value of D independent of z.

It remains to bound from below the value of $|\phi(t)|$ for $t \in \Gamma_1$. For t such that $\operatorname{Re} t \leq -\varepsilon$, we have $|t-\alpha_j| \geq |t-\beta_j|$ for $1 \leq j \leq n$, and so $|\phi(t)| \geq 1$. Then for t such that $-\varepsilon < \operatorname{Re} t \leq 0$, we have $|\phi(t)| \geq |t|/2$, and $|\phi(t)|$ is not small for $t \in \Gamma_1 \cap \{z \in \mathbb{C} : \operatorname{Re} z \leq 0\}$. For $t \in \Gamma_1$ in the right-half plane, $|\phi(t)|$ is root-exponentially small. Fortunately, we can control the rate at which $|\phi(t)|$ decays using the parameter ρ. For any choice of $\rho \in (0,1)$, we can discard α_j and β_j for all j such that $\alpha_j \geq \rho$ from the interpolants. The remaining terms $|t-\alpha_j|/|t-\beta_j|$ are all bounded below by $1-\mathcal{O}(\rho)$. However, since both the α_j and β_j cluster according to (2.28) and (2.29) these terms also approach 1 geometrically as a function of $j/\sqrt{n}$. As such, only $\mathcal{O}(\sqrt{n})$ terms contribute to the product, and $|\phi(t)|$ can be bounded from below by a term that grows like $(1-\mathcal{O}(\rho))^{\sqrt{n}}$. For ρ sufficiently small, $|\phi(t)|$ decays at a root-exponential rate slower than $|\phi(z)|$ as $n \to \infty$. Combining this with (2.35) shows us that $|I_1|$ also asymptotically decays root-exponentially, and therefore (2.26) holds as $n \to \infty$. □

Theorem 2.6 shows that Theorem 2.5 can be extended to the case where the function that is to be approximated has multiple singularities. We first presented this result as Theorem 2.3 in [68].

Theorem 2.6. *Let Ω be a convex polygon with vertices at $\{\zeta_k\}_{k=1}^p$, and let f be an analytic function on Ω that can be extended past each side of Ω and into a disk, slit along the exterior bisector at each of the corners. Further assume that there exists $\delta > 0$ such that $f(z) - f(\zeta_k) = \mathcal{O}(|z-\zeta_k|^\delta)$ as $z \to \zeta_k$ for each $k = 1, 2, \ldots, p$. Then there exist a constant $C > 0$ and degree n rational functions $\{r_n\}_{n=1}^\infty$ such that*

$$\max_{z \in \Omega} |f(z) - r_n(z)| = \mathcal{O}(e^{-C\sqrt{n}}) \tag{2.36}$$

as $n \to \infty$. Moreover, r_n can be taken to have poles clustering at each ζ_k, placed along the exterior bisector, with the same spacing as (2.28) for any $\sigma > 0$, as long as there are at least $\mathcal{O}(n)$ poles near each ζ_k as $n \to \infty$.

Proof. Under the analyticity assumptions, we can express f as a Cauchy integral

$$f(z) = \frac{1}{2\pi i} \int_\Gamma \frac{f(t)}{t-z}\, dt \tag{2.37}$$

where Γ is a contour enclosing Ω which has slits at each vertex and connects the slits with straight sides (see Fig. 2.4). We can split the integral in (2.38) such that

$$f(z) = \frac{1}{2\pi i} \left[\sum_{k=1}^{p} \underbrace{\int_{\Gamma_k} \frac{f(t)}{t-z}\, dt}_{=f_k} + \underbrace{\int_{\Gamma'} \frac{f(t)}{t-z}\, dt}_{=g} \right] \tag{2.38}$$

where Γ_k corresponds to the two sides of the slit part of Γ near the vertex ζ_k for $k = 1, 2, \ldots, p$ and $\Gamma' = \Gamma \setminus \cup_{k=1}^{p} \Gamma_k$. We will show that each term in (2.38) can be approximated by rational functions with root-exponential convergence with respect to the degree.

We start with the term g in (2.38), which corresponds to an integral over Γ'. We note that this function is analytic in an open region containing $\overline{\Omega}$, and it follows from Runge's theorem that we can approximate it on Ω by a polynomial with exponential convergence with respect to the degree [60, Ch. 3]. We now turn our attention to an arbitrary f_k for $1 \le k \le p$ in (2.38). We note that f_k is analytic in a slit disk containing Ω that is a copy of the slit unit disk considered in Theorem 2.5, rotated, translated, and scaled by some arbitrary constant $S > \operatorname{diam} \Omega$. By subtracting out a constant, which can then be absorbed into the approximation of g, we can also assume that $f_k(z) = \mathcal{O}(|z - \zeta_k|^\delta)$ as $z \to \zeta_k$. Then for any choice of $S > \operatorname{diam} \Omega / \rho$ where ρ is the parameter in the statement of Theorem 2.5, it follows that we can find a rational approximation to f_k on Ω that converges root-exponentially with respect to the degree. Summing the rational approximations to each f_k and the polynomial approximation to g then yields a rational function satisfying (2.36). □

Remark 2.5. The theory here has been developed for the specific case of convex polygons, due to our choice of interpolation points in the proof of Theorem 2.5, but we believe that it can be extended to a much more general class of domains with corners. The same strategy of using the Hermite integral formula to obtain a locally accurate approximation to a function with a single singularity, and then decomposing a function with multiple singularities with Cauchy integrals should still apply for domains with reentrant corners. However, in the former step it does not suffice to use interpolation points like (2.29). Nonetheless, we expect that this difficulty can be

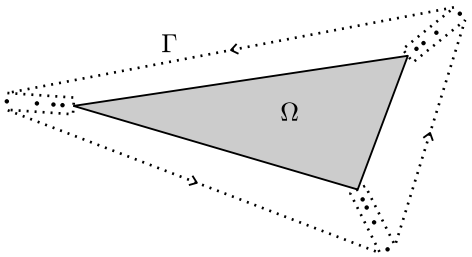


Figure 2.4: Example of a contour used in the proof of Theorem 2.6 in a case when Ω is a triangle

circumvented by placing interpolation points instead on the boundary of Ω, although the details get substantially more cumbersome. Furthermore, nothing in our proofs depends on the sides of Ω being straight.

2.4 Helmholtz solver

2.4.1 Algorithm

Our Laplace solver was motivated by rational approximation theory, which is not directly applicable to the Helmholtz equation. Instead, we derive our Helmholtz solver by means of analogy to the Laplace case. To this end, we recall that a function ϕ that is harmonic outside of the closure of some disk $D = \{z \in \mathbb{C} : |z - z_*| < R\}$ for some $z_* \in \mathbb{C}$ and $R > 0$, and satisfies a logarithmic growth condition at ∞ can be expressed as a multipole expansion of the form

$$\phi(z) = \mathrm{Re}\left[a_0 \log(z - z_*) + \sum_{j=1}^{\infty} \frac{a_j}{(z - z_*)^j}\right], \quad z \in \mathbb{C}\backslash\overline{D} \tag{2.39}$$

for some $\{a_j\}_{j=0}^{\infty} \subset \mathbb{C}$ (see, e.g., [72, 129, 162]). From this perspective the central observation in Section 2.3 is that corner singularities in solutions to the Laplace equation can be resolved by exponentially clustering dipoles (i.e., the $j = 1$ term in the summation in (2.39)) at the singularities. Multipole expansions generalize straightforwardly to the Helmholtz equation. In particular, suppose now that ϕ satisfies the Helmholtz equation (2.1) outside of the closure of D. Then ϕ may be expressed as an expansion of the form

$$\phi(z) = \sum_{j=-\infty}^{\infty} a_j H_j(\kappa|z - z_*|) e^{ij \arg(z - z_*)}, \quad z \in \mathbb{C}\backslash\overline{D} \tag{2.40}$$

for some $\{a_j\}_{j=-\infty}^{\infty} \subset \mathbb{C}$ [102, 130]. In (2.40) H_j denotes the Hankel function of the first kind of order j, which is given by

$$H_j(z) = J_j(z) + iY_j(z), \quad z \in \mathbb{C}\backslash(-\infty, 0] \tag{2.41}$$

for $j \in \mathbb{Z}$ where J_j and Y_j are the Bessel functions of the first and second kinds of order j, respectively, which in turn arise as solutions of Bessel's differential equation. For a comprehensive tabulation of the properties of Hankel and Bessel functions, we refer the reader to [49, Ch. 10]. It turns out that if we exponentially cluster dipoles ($j = \pm 1$ terms in (2.40)) with the same spacing (2.17) we can resolve singularities in a solution of the Helmholtz equation in a manner that still converges root-exponentially with respect to the number of degrees of freedom. This can also be understood from the asymptotic

$$H_\nu(z) \sim -\frac{i}{\pi}\Gamma(\nu)\left(\frac{1}{2}z\right)^{-\nu}, \quad z \to 0 \tag{2.42}$$

which holds for any ν with $\operatorname{Re}\nu > 0$ and where Γ is the gamma function [49, 10.7.7], paired with the identity

$$H_{-j}(z) = (-1)^j H_j(z), \quad z \in \mathbb{C}\backslash(-\infty, 0] \tag{2.43}$$

which holds for any $j \in \mathbb{Z}$ [49, 10.4.2]. We see from (2.42) and (2.43) that the dipole terms in (2.40) are the terms in the expansion with the same simple pole singularities as in the Laplace case.

To derive the analog of the polynomial term for the smooth part of the solution in (2.16), we use the well-known fact that if a function ϕ satisfies the Helmholtz equation in the aforementioned disk D then ϕ may be expanded as

$$\phi(z) = \sum_{j=-\infty}^{\infty} b_j J_j(\kappa|z - z_*|)e^{ij\arg(z-z_*)}, \quad z \in D \tag{2.44}$$

for some $\{b_j\}_{j=-\infty}^{\infty} \subset \mathbb{C}$ [102, 130]. The expansion (2.44) is smooth in the plane but does not satisfy the Sommerfeld radiation condition. Therefore, for exterior problems we instead need to use an expansion of the form (2.40).

In summary, the correct analog of (2.16) is

$$u(z) \approx \sum_{j=1}^{N_1} a_j H_1(\kappa|z - z_j|)e^{i\arg(z-z_j)} + b_j H_{-1}(\kappa|z - z_j|)e^{-i\arg(z-z_j)} + \sum_{j=-N_2}^{N_2} c_j J_j(\kappa|z - z_*|)e^{ij\arg(z-z_*)} \tag{2.45}$$

where $\{z_j\}_{j=1}^{N_1} \subset \mathbb{C}\backslash\overline{\Omega}$ and $z_* \in \Omega$ are still fixed a priori, and now $\{a_j\}_{j=1}^{N_1} \subset \mathbb{C}$, $\{b_j\}_{j=1}^{N_1} \subset \mathbb{C}$, and $\{c_j\}_{j=-N_2}^{N_2} \subset \mathbb{C}$ will be determined by least-squares fitting. Naturally, the maximum principle will depend on the conditioning of the problem; if u and v satisfy the homogeneous Helmholtz equation with wavenumber κ in the interior of Ω, then we are only guaranteed that

$$\|u - v\|_{L^2(\Omega)} \leq \frac{C_\Omega}{d_{\kappa,\Omega}} \|u - v\|_{L^2(\Gamma)} \tag{2.46}$$

where $C_\Omega > 0$ is a constant that depends on the domain and $d_{\kappa,\Omega} = \min_j |\kappa^2 - \lambda_j^2|/|\lambda_j|$ where $\{\lambda_j\}_{j=1}^{\infty}$ are the Dirichlet eigenvalues of the Laplacian on Ω [14].

The analog of (2.25), which is the relevant ansatz for (2.1), is given by

$$u(z) \approx \sum_{j=1}^{N_1} a_j H_1(\kappa|z - z_j|)e^{i\arg(z-z_j)} + b_j H_{-1}(\kappa|z - z_j|)e^{-i\arg(z-z_j)} + \sum_{j=-N_2}^{N_2} c_j H_j(\kappa|z - z_*|)e^{ij\arg(z-z_*)} \tag{2.47}$$

where now we will take $\{z_j\}_{j=1}^{N_1} \subset \Omega$ and $z_* \in \Omega$. Note that in (2.47) u is guaranteed to satisfy the Sommerfeld radiation condition since each term in the expansion satisfies it. In this case the constant in the right-hand side of the bound analogous to (2.46) will depend on the nearest complex-valued exterior resonance to κ, instead of the nearest interior eigenvalue.

The procedure described in Section 2.3.1 can now easily be appropriately modified. The $\{z_j\}_{j=1}^{N_1}$ can still be chosen according to (2.17), and Γ can be discretized in the same fashion as for the Laplace case. The coefficient matrix must be modified to account for the new ansatzes (2.45) and (2.47). For the exterior problem, for example, the entries of $\mathbf{A} \in \mathbb{C}^{M\times N}$ are now given by

$$\mathbf{A}(i,j) = \begin{cases} H_1(\kappa r_{i,j})e^{i\theta_{i,j}}, & 1 \leq j \leq N_1, \\ H_{-1}(\kappa r_{i,j-N_1})e^{-i\theta_{i,j-N_1}}, & N_1 + 1 \leq j \leq 2N_1, \\ H_{j-2N_1-1}(\kappa r_{i,*})e^{i(j-2N_1-1)\theta_{i,*}}, & 2N_1 + 1 \leq j \leq 2N_1 + N_2 + 1, \\ H_{-j+2N_1+N_2+1}(\kappa r_{i,*})e^{-i(j-2N_1-N_2-1)\theta_{i,*}}, & 2N_1 + N_2 + 2 \leq j \leq N \end{cases} \tag{2.48}$$

where $r_{i,j} = |w_i - z_j|$, $\theta_{i,j} = \arg(w_i - z_j)$, $r_{i,*} = |w_i - z_*|$, and $\theta_{i,*} = \arg(w_i - z_*)$, and the corresponding coefficient vector $\mathbf{c} \in \mathbb{C}^N$ has entries

$$\mathbf{c}(j) = \begin{cases} a_j, & 1 \leq j \leq N_1, \\ b_{j-N_1}, & N_1 + 1 \leq j \leq 2N_1, \\ c_{j-2N_1-1}, & 2N_1 + 1 \leq j \leq 2N_1 + N_2 + 1, \\ c_{-j+2N_1+N_2+1}, & 2N_1 + N_2 + 2 \leq j \leq N \end{cases} \tag{2.49}$$

for $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$. For interior problems, the Hankel functions in the bottom two cases in (2.48) are simply replaced by the corresponding Bessel functions of the first kind. Pseudocode for this procedure for exterior problems is provided in Algorithm 2.2. After the solution is computed, it can be evaluated at any collection of points in $\{t_i\}_{i=1}^{P} \subset \mathbb{C}$ by forming the matrix $\mathbf{A}_{t_i,z_j,z_*} \in \mathbb{C}^{P\times N}$ with entries given by the right-hand side (2.48) where now $r_{i,j} = |t_i - z_j|$, $\theta_{i,j} = \arg(t_i - z_j)$, $r_{i,*} = |t_i - z_*|$, and $\theta_{i,*} = \arg(t_i - z_*)$ and again evaluating $\mathbf{A}_{t_i,z_j,z_*}\mathbf{c}$.

Algorithm 2.2 Solver for planar exterior, sound-soft Helmholtz problems

Input: Boundary Γ with corners at $\{\zeta\}_{k=1}^{p}$, boundary data h, center of multipole expansion z_*, and convergence tolerance $\varepsilon > 0$

Output: Dipole locations $\{z_j\}_{j=1}^{N_1}$ and coefficients $\{a_j\}_{j=1}^{N_1}$, $\{b_j\}_{j=1}^{N_1}$, and $\{c_j\}_{j=-N_2}^{N_2}$ (2.47)

for all n increasing quadratically **do**
  Generate dipole locations $\{z_j\}_{j=1}^{N_1} \subset \Omega$ where the points cluster at $\{\zeta_k\}_{k=1}^{p}$ with spacing (2.17) and $N_1 = \mathcal{O}(pn)$
  Fix $N_2 = \mathcal{O}(n)$, and set $N = 2N_1 + 2N_2 + 1$
  Generate a sample set $\{w_i\}_{i=1}^{M} \subset \Gamma$ that also exponentially clusters at $\{\zeta_k\}_{k=1}^{p}$ where $M = \mathcal{O}(N)$
  Assemble $\mathbf{A} \in \mathbb{C}^{M\times N}$ according to (2.48) and $\mathbf{b} \in \mathbb{C}^{M}$ according to (2.18)
  Form $\mathbf{D}$ in (2.22)
  Compute $\widetilde{\mathbf{c}} = \arg\min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{A}\mathbf{D}^{-1}\mathbf{x} - \mathbf{b}\|$ using a backward stable algorithm
  Evaluate $\mathbf{c} = \mathbf{D}\widetilde{\mathbf{c}}$
  if $\|\mathbf{A}\mathbf{c} - \mathbf{b}\|_\infty < \varepsilon$ **then**
    Obtain $\{a_j\}_{j=1}^{N_1}$, $\{b_j\}_{j=1}^{N_1}$, and $\{c_j\}_{j=-N_2}^{N_2}$ from $\mathbf{c}$ using (2.49)
    return $\{z_j\}_{j=1}^{N_1}$, $\{a_j\}_{j=1}^{N_1}$, $\{b_j\}_{j=1}^{N_1}$, and $\{c_j\}_{j=-N_2}^{N_2}$
  end if
end for

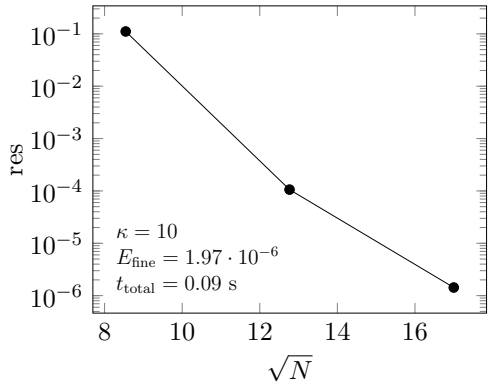
Remark 2.6. It is clear that the same complexity analysis for the Laplace case in Remark 2.2 holds for the Helmholtz solver, as well. However, the constants in the complexity estimates are substantially worse. This is partially due to the increased cost associated with complex arithmetic and evaluating the Hankel and Bessel functions during the assembly of the coefficient matrix. Additionally, as the wavenumber of the problem grows, $\mathcal{O}(\kappa)$ terms are needed just to resolve the oscillations in the solution and therefore achieve an error smaller than order 1. As such, we believe this method is not viable in the high-frequency regime.

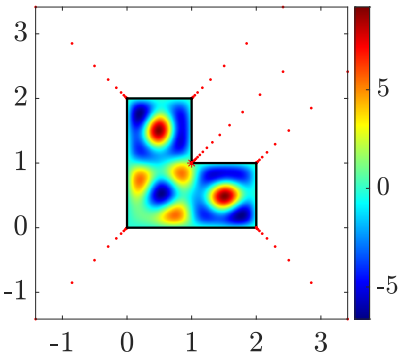
2.4.2 Numerical experiments

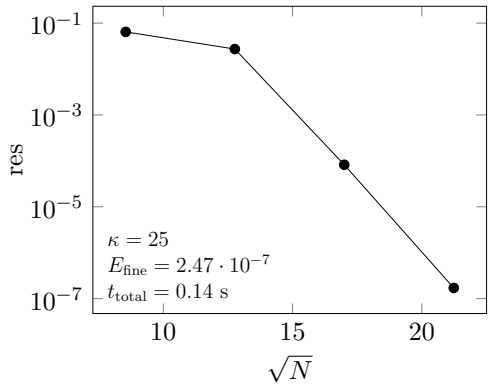
We will illustrate our solver on several sound-soft Helmholtz problems. We summarize our results for each problem in a row of Fig. 2.5 with two plots, with roughly the same convention as Section 2.3.2. A notable difference is that we now plot the total field, instead of just the computed scattered field, in the images on the right (i.e., we plot $u + u_{\text{inc}}$). The red dots still denote the $\{z_j\}_{j=1}^{N_1}$ at the final iteration, where now these are the locations of the dipoles. Similarly, the red asterisk denotes z_*, the center of a truncated expansion of the form (2.44) for the interior problem and (2.40) for the exterior problems. Again, an a posteriori error estimate (E_{fine}), given by the residual evaluated on a mesh four-times as fine as the one used for the final least-squares solution, and total computation time (t_{total}) are displayed in the bottom left corners of the plots on the left. The problems considered here are:

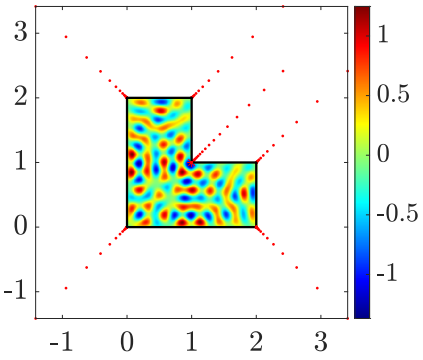
L-shaped domain	These are interior sound-soft problems on the L-shaped domain. The incident fields are given by the point source $u_{\text{inc}}(z) = Y_0(\kappa\|z - 0.5 - 0.5i\|)$, and the convergence tolerance is set to 10^{-5}. We consider cases when the wavenumbers are given by $\kappa = 10, 25, 50, 200$ (so that $\operatorname{diam}\Omega \approx 3.2, 8.0, 15.9, 63.7\,\lambda$).
Five-pointed star	These are exterior, sound-soft scattering problems where the scatterer is the five-pointed star. We consider the case when $\kappa = 25$ ($\operatorname{diam}\Omega \approx 7.6\,\lambda$) and the incident field is given by the point source $u_{\text{inc}}(z) = H_0(\kappa\|z-1-i\|)$, and the case when $\kappa = 50$ ($\operatorname{diam}\Omega \approx 15.1\,\lambda$) and the incident field is given by the plane wave $u_{\text{inc}}(z) = e^{i\kappa(\operatorname{Re}[z]-\operatorname{Im}[z])/\sqrt{2}}$. The convergence tolerance is set to 10^{-5}.
United Kingdom	This is an exterior, sound-soft scattering problem where the scatterer is the model of the United Kingdom. We take $\kappa = 25$ ($\operatorname{diam}\Omega \approx 7.2\,\lambda$) and the incident field to be the plane wave $u_{\text{inc}}(z) = e^{i\kappa \operatorname{Re}[z]}$. The convergence tolerance is set to 10^{-5}.
Plane	This is an exterior, sound-soft scattering problems where the scatterer is given by the plane domain and $\kappa = 10$ ($\operatorname{diam}\Omega \approx 19.3\,\lambda$). We consider the case when the incident field is the plane wave $u_{\text{inc}}(z) = e^{i\kappa(\operatorname{Re}[z]-\operatorname{Im}[z])/\sqrt{2}}$. The convergence tolerance is set to $2 \cdot 10^{-5}$.[5]

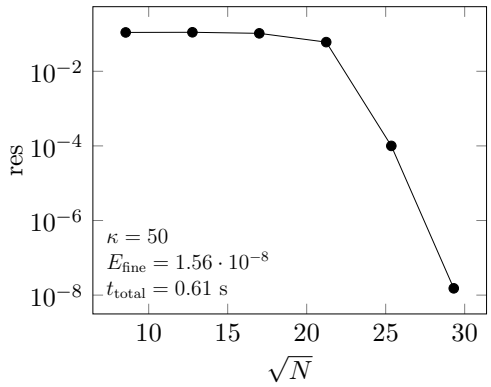
[5]Due to the elongated nature of this domain, we get stagnation after the residual falls below $1.67 \cdot 10^{-5}$ for these problems. This could be remedied by replacing the terms in the expansion in (2.47) with a better-conditioned basis. See Remark 2.4.

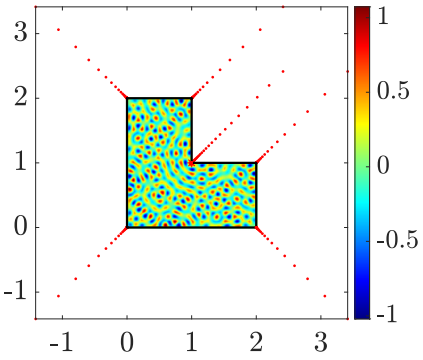

res
$\sqrt{N}$
$\kappa = 10$
$E_{\text{fine}} = 1.97 \cdot 10^{-6}$
$t_{\text{total}} = 0.09$ s

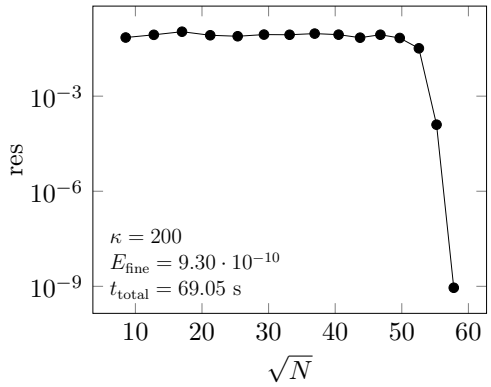


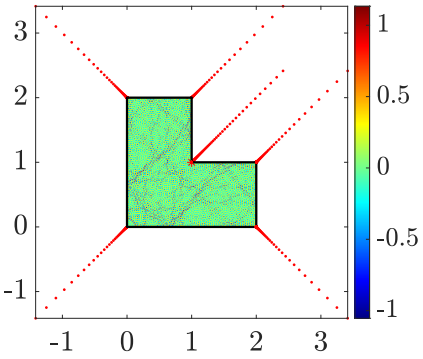

res
$\sqrt{N}$
$\kappa = 25$
$E_{\text{fine}} = 2.47 \cdot 10^{-7}$
$t_{\text{total}} = 0.14$ s




res
$\sqrt{N}$
$\kappa = 50$
$E_{\text{fine}} = 1.56 \cdot 10^{-8}$
$t_{\text{total}} = 0.61$ s




res
$\sqrt{N}$
$\kappa = 200$
$E_{\text{fine}} = 9.30 \cdot 10^{-10}$
$t_{\text{total}} = 69.05$ s



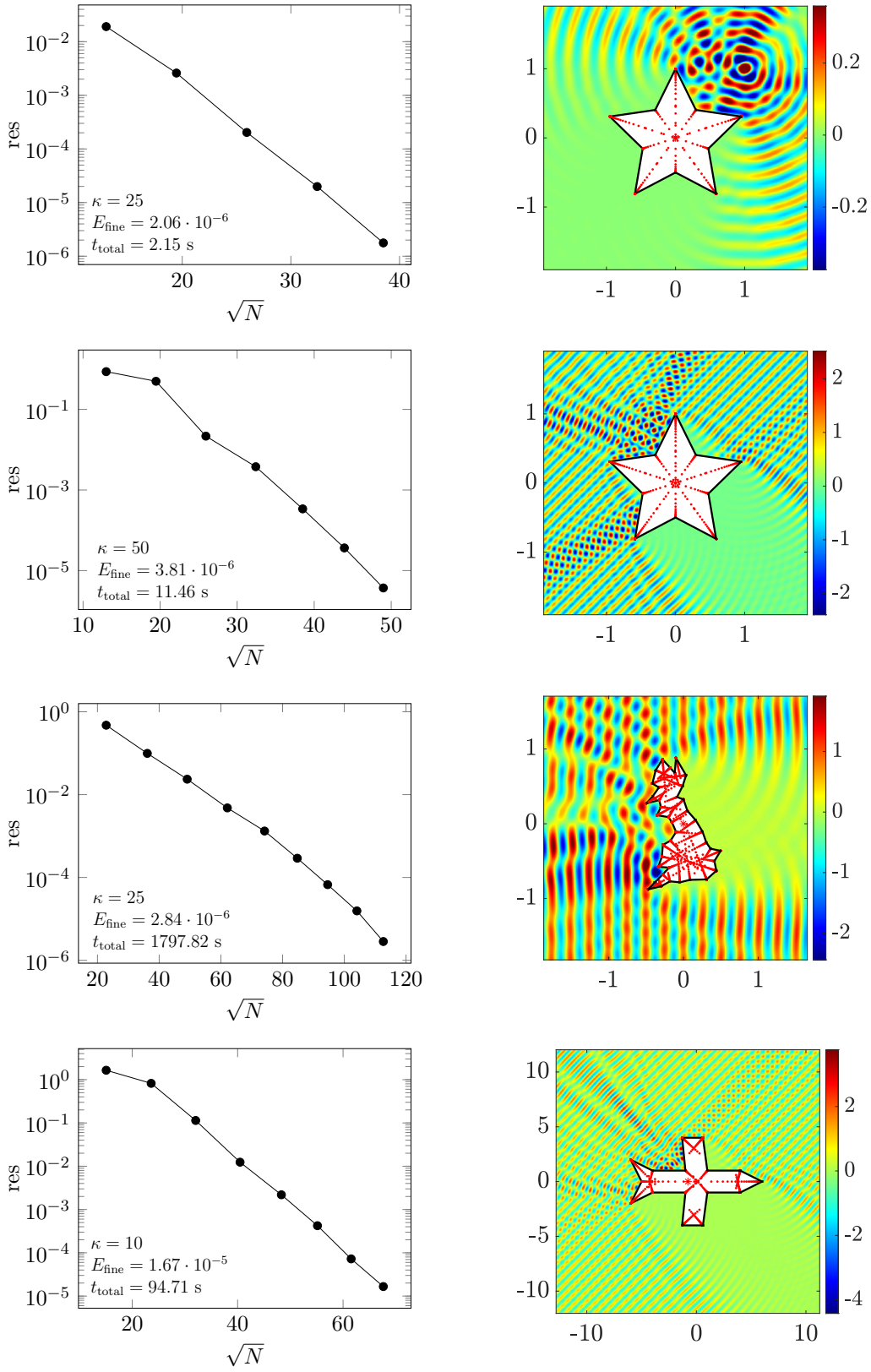


Figure 2.5: Plots for Section 2.4.2

2.4.3 Variants

Since we are using a global representation of the solution that exactly satisfies the Helmholtz equation, our solver is compatible with a broad range of techniques, often introduced in the context of boundary integral equation methods or the MFS, to handle variants of the problem (2.1). Here, we will consider exterior multibody problems, rough surface scattering, sound-hard scattering, and transmission problems, but other variants are also possible.

Multibody problems

A simple variant of (2.1) is the case when there are multiple disjoint scatterers. For simplicity we will restrict ourselves to the case where there are two scatterers Ω_1 and Ω_2 with boundaries Γ_1 and Γ_2, respectively, but the case with three or more scatterers is entirely straightforward. In this case we will choose $\{z_j^{(1)}\}_{j=1}^{N_1^{(1)}}$ and $\{z_j^{(2)}\}_{j=1}^{N_1^{(2)}}$ that cluster exponentially at the corners of Ω_1 and Ω_2, respectively, and $z_*^{(1)}$ and $z_*^{(2)}$ centered in the respective domain. We then use an ansatz of the form

$$\begin{aligned} u(z) \approx &\sum_{j=1}^{N_1^{(1)}} a_j^{(1)} H_1(\kappa|z - z_j^{(1)}|) e^{i \arg(z - z_j^{(1)})} + \sum_{j=1}^{N_1^{(1)}} b_j^{(1)} H_{-1}(\kappa|z - z_j^{(1)}|) e^{-i \arg(z - z_j^{(1)})} \\ &+ \sum_{j=-N_2^{(1)}}^{N_2^{(1)}} c_j^{(1)} H_j(\kappa|z - z_*^{(1)}|) e^{ij \arg(z - z_*^{(1)})} + \sum_{j=1}^{N_1^{(2)}} a_j^{(2)} H_1(\kappa|z - z_j^{(2)}|) e^{i \arg(z - z_j^{(2)})} \\ &+ \sum_{j=1}^{N_1^{(2)}} b_j^{(2)} H_{-1}(\kappa|z - z_j^{(2)}|) e^{-i \arg(z - z_j^{(2)})} + \sum_{j=-N_2^{(2)}}^{N_2^{(2)}} c_j^{(2)} H_j(\kappa|z - z_*^{(2)}|) e^{ij \arg(z - z_*^{(2)})}. \end{aligned} \tag{2.50}$$

Now let $\{w_i^{(1)}\}_{i=1}^{M^{(1)}}$ and $\{w_i^{(2)}\}_{i=1}^{M^{(2)}}$ be discretizations of Γ_1 and Γ_2, respectively. The coefficients $\{a_j^{(1)}\}_{j=1}^{N_1^{(1)}}$, $\{b_j^{(1)}\}_{j=1}^{N_1^{(1)}}$, $\{c_j^{(1)}\}_{j=-N_2^{(1)}}^{N_2^{(1)}}$, $\{a_j^{(2)}\}_{j=1}^{N_1^{(2)}}$,$\{b_j^{(2)}\}_{j=1}^{N_1^{(2)}}$, and $\{c_j^{(2)}\}_{j=-N_2^{(2)}}^{N_2^{(2)}}$ in (2.50) can be found by solving a least-squares problem where the coefficient matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$ where $M = M^{(1)} + M^{(2)}$ and $N = 2N_1^{(1)} + 2N_1^{(2)} + 2N_2^{(1)} + 2N_2^{(2)} + 2$ is now given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{w_i^{(1)}, z_j^{(1)}, z_*^{(1)}} & \mathbf{A}_{w_i^{(1)}, z_j^{(2)}, z_*^{(2)}} \\ \mathbf{A}_{w_i^{(2)}, z_j^{(1)}, z_*^{(1)}} & \mathbf{A}_{w_i^{(2)}, z_j^{(2)}, z_*^{(2)}} \end{bmatrix} \tag{2.51}$$

and the entries of the right-hand side $\mathbf{b} \in \mathbb{C}^M$ are given by

$$\mathbf{b}(i) = \begin{cases} -u_{\text{inc}}(w_i^{(1)}), & 1 \le i \le M^{(1)}, \\ -u_{\text{inc}}(w_{i-M^{(1)}}^{(2)}), & M^{(1)} + 1 \le i \le M. \end{cases} \tag{2.52}$$

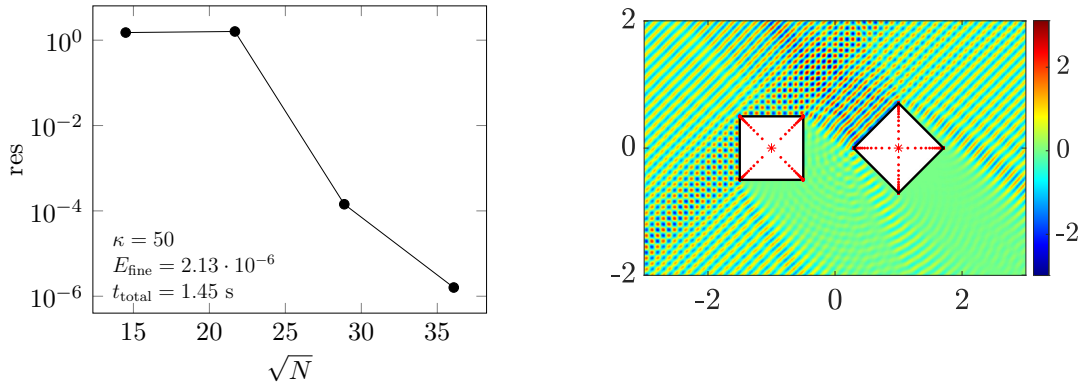


Figure 2.6: Convergence and total field plots for an exterior, sound-soft scattering problem where the scatterers are two squares and $\kappa = 50$

A numerical experiment where a plane wave $u_{\text{inc}}(z) = e^{i\kappa(\text{Re}[z]-\text{Im}[z])/\sqrt{2}}$ with $\kappa = 50$ impinges on two squares is displayed in Fig. 2.6.

Rough surface scattering

In the rough surface scattering problem, the scattered field generated by a point source impinging on a locally perturbed half-space is computed. Specifically, we solve

$$\begin{cases} -\Delta u(z) - \kappa^2 u(z) = 0, & z \text{ above } \Gamma, \\ u(z) = -u_{\text{inc}}(z), & z \in \Gamma, \\ \sqrt{r}|\partial_r u(z) - i\kappa u(z)| = 0, & r = |z| \to \infty \end{cases} \tag{2.53}$$

where Γ is a continuous, piecewise analytic curve that can be parametrized by an injective function $\gamma : (-\infty, \infty) \to \mathbb{C}$ where there exists $T > 0$ such that $\text{Im}[\gamma(t)] = 0$ for $|t| \geq T$ and $\text{Im}[\gamma(t)] \geq 0$ for $t \in (-T, T)$. We let $\widehat{\Gamma} = \{\gamma(t) : t \in [-T, T]\}$. An example of such a curve is given in Fig. 2.7(a).

The incident field is typically given by a point source of the form

$$u_{\text{inc}}(z) = qH_0(\kappa|z - \zeta|) \tag{2.54}$$

for some ζ above Γ and $q \in \mathbb{C}$. In the event we have a half-space problem (so that $\Gamma = \mathbb{R}$), the method of images shows that $u(z) = -qH_0(\kappa|z - \overline{\zeta}|)$ solves (2.53). Thus, for general Γ, we can express $u(z) = \widehat{u}(z) - qH_0(\kappa|z - \overline{\zeta}|)$ where $\widehat{u}$ satisfies

$$\begin{cases} -\Delta\widehat{u}(z) - \kappa^2\widehat{u}(z) = 0, & z \text{ above } \Gamma, \\ \widehat{u}(z) = qH_0(\kappa|z - \overline{\zeta}|) - qH_0(\kappa|z - \zeta|), & z \in \widehat{\Gamma}, \\ \widehat{u}(z) = 0, & z \in \Gamma \backslash \widehat{\Gamma}, \\ \sqrt{r}|\partial_r\widehat{u}(z) - i\kappa\widehat{u}(z)| = 0, & r = |z| \to \infty. \end{cases} \tag{2.55}$$

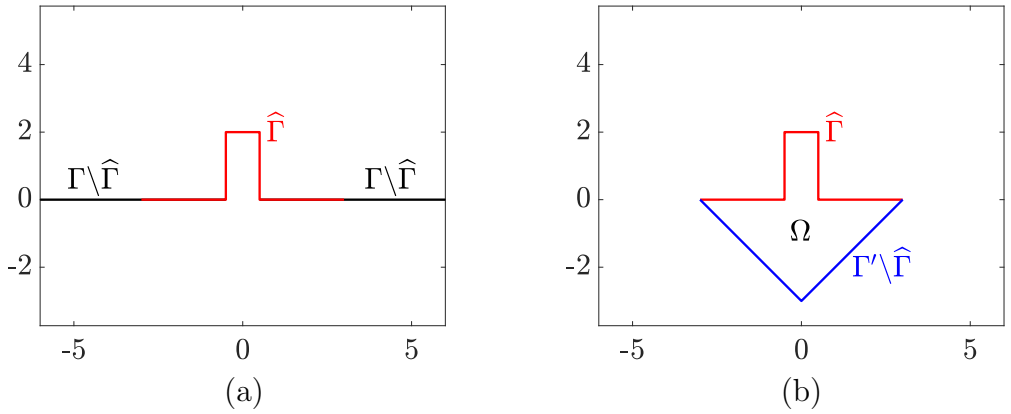


Figure 2.7: (a) An example of the geometry in a rough surface scattering problem. The scattered field must satisfy the Helmholtz equation above the curve and negate the incident field on the boundary. (b) A bounded scatterer with which the problem can be reformulated using the approach of [174].

There exist several different methods of solving (2.55) in the boundary integral equations literature (see, e.g., [6, 104, 105, 127, 164, 174]). We will take an approach introduced by Zhang and Zhang in [174]. There, they showed that $\widehat{u}$ could be obtained by instead solving the exterior problem

$$\begin{cases} -\Delta\widehat{u}(z) - \kappa^2\widehat{u}(z) = 0, & z \in \mathbb{C}\backslash\overline{\Omega}, \\ \widehat{u}(z) = qH_0(\kappa|z-\overline{\zeta}|) - qH_0(\kappa|z-\zeta|), & z \in \widehat{\Gamma}, \\ \widehat{u}(z) + \widehat{u}(\overline{z}) = 0, & z \in \Gamma'\backslash\widehat{\Gamma}, \\ \sqrt{r}|\partial_r\widehat{u}(z) - i\kappa\widehat{u}(z)| = 0, & r = |z| \to \infty \end{cases} \tag{2.56}$$

where Ω is a simply connected domain with piecewise analytic boundary Γ' where $\widehat{\Gamma} \subset \Gamma'$, $\Gamma'\backslash\widehat{\Gamma} \subset \{z : \operatorname{Im} z \leq 0\}$, and $\{\overline{z} : z \in \widehat{\Gamma}\} \subset \Omega$. Such an Ω for the geometry in Fig. 2.7(a) is given in Fig. 2.7(b). While (2.56), unlike (2.55), is posed with a bounded scatterer, this comes at the cost of introducing corners in the domain at $\gamma(T)$ and $\gamma(-T)$. Fortunately, these can be handled with our solver by using an ansatz of the form (2.47) for $\widehat{u}$. We let $\{z_j\}_{j=1}^{N_1}$ be dipole locations that cluster exponentially at the corners of Ω, z_* be centered in Ω, and $\{w_i\}_{i=1}^{M_1}$ and $\{w_i'\}_{i=1}^{M_2}$ be discretizations of $\widehat{\Gamma}$ and $\Gamma'\backslash\widehat{\Gamma}$, respectively. The relevant coefficient matrix $\mathbf{A} \in \mathbb{C}^{M\times N}$ where $M = M_1 + M_2$ and $N = 2N_1 + 2N_2 + 1$ is then given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{w_i,z_j,z_*} \\ \mathbf{A}_{w_i',z_j,z_*} + \mathbf{A}_{\overline{w_i'},z_j,z_*} \end{bmatrix} \tag{2.57}$$

and the entries of the right-hand side are given by

$$\mathbf{b}(i) = \begin{cases} qH_0(\kappa|w_i-\overline{\zeta}|) - qH_0(\kappa|w_i-\zeta|), & 1 \leq i \leq M_1, \\ 0, & M_1+1 \leq i \leq M \end{cases} \tag{2.58}$$

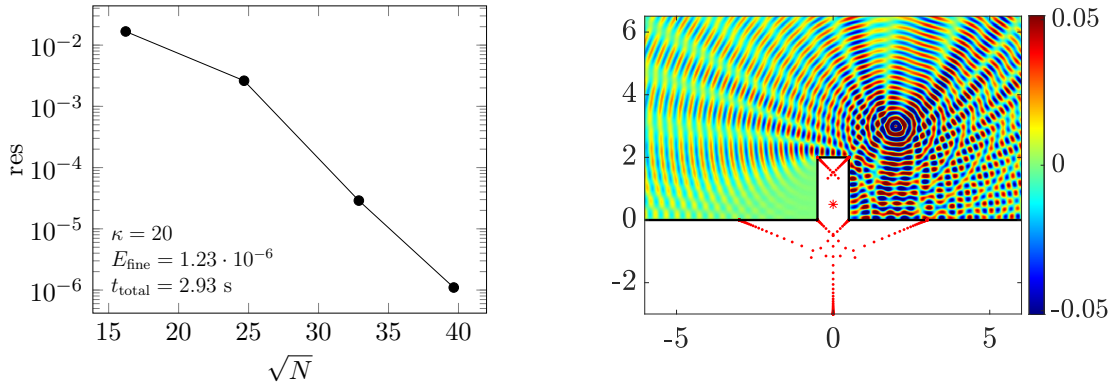


Figure 2.8: Convergence and total field plots for a sound-soft rough surface scattering problem

for $i = 1, 2, \ldots, M$. The result when this procedure is applied to the geometry given in Fig. 2.7(a) when $\kappa = 20$ and the incident field is given by $u_{\text{inc}}(z) = \frac{i}{4}H_0(\kappa|z - 2 - 3i|)$ is shown in Fig. 2.8.

Sound-hard scattering

We can also easily modify our solver to handle scattering from a sound-hard obstacle, which corresponds to the presence of a Neumann boundary condition. The sound-hard analog of (2.1) is given by

$$\begin{cases} -\Delta u(z) - \kappa^2 u(z) = 0, & z \in \mathbb{C}\backslash\overline{\Omega}, \\ [\partial_n u](z) = -[\partial_n u_{\text{inc}}](z), & z \in \Gamma, \\ \sqrt{r}|\partial_r u(z) - i\kappa u(z)| = 0, & r = |z| \to \infty \end{cases} \tag{2.59}$$

where ∂_n is the derivative in the direction that is outward normal to Γ at z. Using the identity

$$\frac{d}{dz}H_j(z) = -H_{j+1}(z) + \frac{j}{z}H_j(z) \tag{2.60}$$

which holds for any $j \in \mathbb{Z}$ [49, 10.6.2], we can derive the identities

$$\begin{aligned} \frac{\partial}{\partial x}\left[H_j(\kappa r)e^{ij\theta}\right] &= \left[-\kappa\cos(\theta)H_{j+1}(\kappa r) + \frac{j\cos(\theta)}{r}H_j(\kappa r) - \frac{ij\sin(\theta)}{r}H_j(\kappa r)\right]e^{ij\theta}, \\ \frac{\partial}{\partial y}\left[H_j(\kappa r)e^{ij\theta}\right] &= \left[-\kappa\sin(\theta)H_{j+1}(\kappa r) + \frac{j\sin(\theta)}{r}H_j(\kappa r) + \frac{ij\cos(\theta)}{r}H_j(\kappa r)\right]e^{ij\theta} \end{aligned} \tag{2.61}$$

for any $j \in \mathbb{Z}$ where $r = |z - \zeta|$ and $\theta = \arg(z - \zeta)$ for $z = x + iy$ and any $\zeta \in \mathbb{C}$. Thus, we can analytically evaluate gradients of (2.47). The relevant coefficient matrix

is now given by

$$\mathbf{A}'(i,j) = \begin{cases} [\partial_n H_1(\kappa r_j) e^{i\theta_j}]_{z=w_i}, & 1 \le j \le N_1, \\ [\partial_n H_{-1}(\kappa r_{j-N_1}) e^{-i\theta_{j-N_1}}]_{z=w_i}, & N_1+1 \le j \le 2N_1, \\ [\partial_n H_{j'}(\kappa r_*) e^{ij'\theta_*}]_{z=w_i}, & 0 \le j' \le N_2, \\ [\partial_n H_{-j'+N_2}(\kappa r_*) e^{-i(j'-N_2)\theta_*}]_{z=w_i}, & N_2+1 \le j' \le 2N_2 \end{cases} \tag{2.62}$$

where $j' = j - 2N_1 - 1$, $r_j = |z - z_j|$, $\theta_j = \arg(z - z_j)$, $r_* = |z - z_*|$, and $\theta_* = \arg(z - z_*)$ for $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$. The entries of the corresponding right-hand side $\mathbf{b} \in \mathbb{C}^M$ are given by $\mathbf{b}(i) = -[\partial_n u_{\text{inc}}](w_i)$ for $1 \le i \le M$.

A technicality now arises. While the solution to (2.59), under our regularity assumptions, will extend continuously to the boundary, its normal derivative will typically be discontinuous at the corners since the normal vectors are undefined there. As such, some care must be exercised when formulating the least-squares problem. Fortunately, we find that it suffices to simply weight the rows of the least-squares problem by the distance of each target point to the closest discontinuity. This can be done by introducing the diagonal matrix $\mathbf{W} \in \mathbb{R}^{M \times M}$ with entries given by

$$\mathbf{W}(i,i) = \min\{|w_i - \zeta_k| : k = 1, 2, \ldots, p\} \tag{2.63}$$

for $i = 1, 2, \ldots, M$ where again $\{\zeta_k\}_{k=1}^p$ are the locations of the corners of Ω. We then solve the least-squares problem

$$\widetilde{\mathbf{c}} = \underset{\mathbf{x} \in \mathbb{C}^N}{\arg\min} \, \|\mathbf{W}\mathbf{A}'\mathbf{D}^{-1}\mathbf{x} - \mathbf{W}\mathbf{b}\| \tag{2.64}$$

where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is now the diagonal matrix that scales the columns of $\mathbf{A}'$ to have unit norms. After this, the vector $\mathbf{c} = \mathbf{D}\widetilde{\mathbf{c}}$ can be computed, and the solution u is then given by (2.47) where the coefficients are still given by (2.49). An error estimate can be obtained by evaluating $\|\mathbf{W}\mathbf{A}'\mathbf{c} - \mathbf{W}\mathbf{b}\|_\infty$. This procedure is illustrated in Fig. 2.9 for the case when the plane wave $u_{\text{inc}}(z) = e^{i\kappa(-\operatorname{Re}[z] - \operatorname{Im}[z])/\sqrt{2}}$ impinges on a sound-hard square when $\kappa = 30$. The more general impedance boundary condition

$$[\partial_n u](z) - i\kappa\beta u(z) = h(z), \quad z \in \Gamma \tag{2.65}$$

can be handled by changing the coefficient matrix to $\mathbf{A}' - i\kappa\beta\mathbf{A}$ and making the appropriate modifications to the right-hand side.

Transmission boundary conditions

In transmission problems the scatterers are penetrable and possess different speeds of

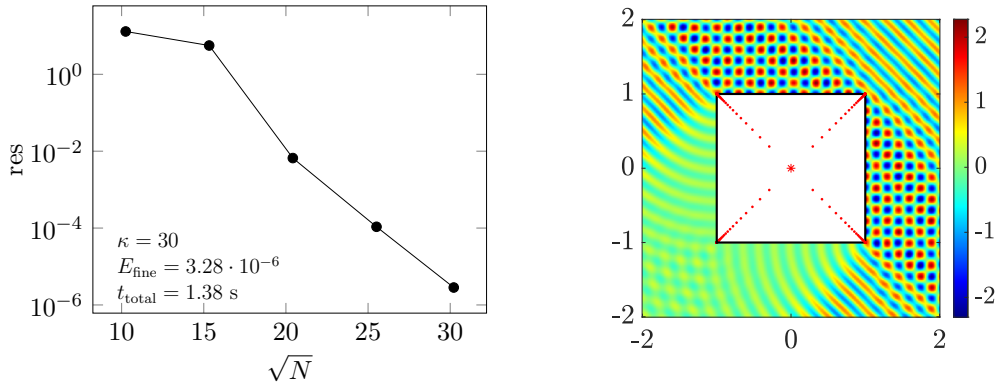


Figure 2.9: Convergence and total field plots for a sound-hard scattering problem where the scatterer is a square

sound than the free space. These situations can be modeled by coupling Helmholtz equations in the interior and exterior with continuity conditions of the form

$$\begin{cases} -\Delta u_+(z) - \kappa_+^2 u_+(z) = 0, & z \in \mathbb{C}\backslash\overline{\Omega}, \\ -\Delta u_-(z) - \kappa_-^2 u_-(z) = 0, & z \in \Omega, \\ u_+(z) - u_-(z) = -u_{\text{inc}}(z), & z \in \Gamma, \\ [\partial_n u_+](z) - [\partial_n u_-](z) = -[\partial_n u_{\text{inc}}](z), & z \in \Gamma, \\ \sqrt{r}|\partial_r u_+(z) - i\kappa u_+(z)| = 0, & r = |z| \to \infty \end{cases} \tag{2.66}$$

where κ_+ and κ_- are the wavenumbers in the exterior and interior, respectively. The resulting total field is then given by

$$u_{\text{tot}}(z) = \begin{cases} u_+(z) + u_{\text{inc}}(z), & z \in \mathbb{C}\backslash\overline{\Omega}, \\ u_-(z), & z \in \overline{\Omega}. \end{cases} \tag{2.67}$$

For these types of problems, we represent u_+ using an ansatz of the form (2.47) where $\kappa = \kappa_+$. Similarly, we represent u_- using an ansatz of the form (2.45) with $\kappa = \kappa_-$. Suppose now that $\{z_j^+\}_{j=1}^{N_1^+} \subset \mathbb{C}\backslash\overline{\Omega}$ and $\{z_j^-\}_{j=1}^{N_1^-} \subset \Omega$ cluster exponentially at the corners of Ω with the appropriate spacing, z_* is centered in Ω, and $\{w_i\}_{i=1}^M$ is a discretization of Γ. The relevant coefficient matrix $\mathbf{A} \in \mathbb{C}^{2M \times N}$ is then given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{w_i, z_j^+, z_*} & -\mathbf{A}_{w_i, z_j^+, z_*} \\ \mathbf{A}'_{w_i, z_j^-, z_*} & -\mathbf{A}'_{w_i, z_j^-, z_*} \end{bmatrix} \tag{2.68}$$

where we understand the entries in the first block column have $\kappa = \kappa_+$ and those in the second block column have $\kappa = \kappa_-$. The right-hand side $\mathbf{b} \in \mathbb{C}^{2M}$ will have entries given by

$$\mathbf{b}(i) = \begin{cases} -u_{\text{inc}}(w_i), & 1 \le i \le M, \\ -[\partial_n u_{\text{inc}}](w_{i-M}), & M+1 \le i \le 2M \end{cases} \tag{2.69}$$

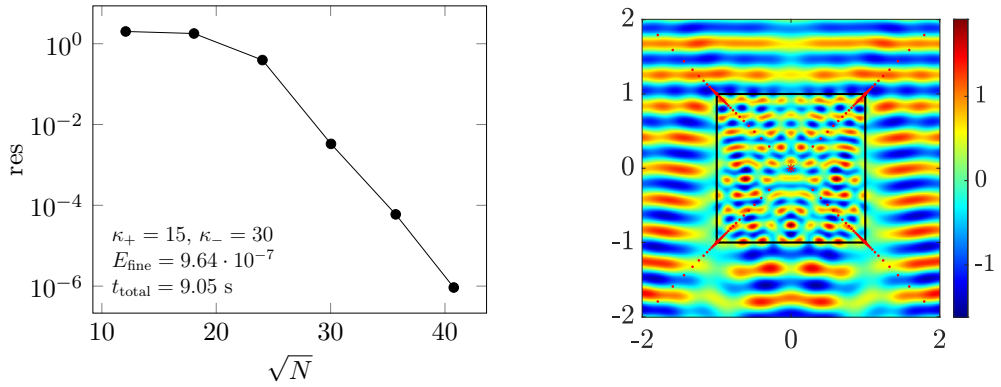


Figure 2.10: Convergence and total field plots for a transmission problem where the scatterer is a square

for $i = 1, 2, \ldots, 2M$, and the entries of the coefficient vector will be ordered so that those for u_+ are first and the remaining entries are the coefficients for u_-. The appropriate weighting matrix $\mathbf{W} \in \mathbb{R}^{2M \times 2M}$ has entries

$$\mathbf{W}(i,i) = \begin{cases} 1, & 1 \le i \le M, \\ \min\{|w_{i-M} - \zeta_k| : k = 1, 2, \ldots, p\}, & M+1 \le i \le 2M. \end{cases} \tag{2.70}$$

We illustrate an example of this in Fig. 2.10 for a transmission problem where the scatterer is a square, $\kappa_- = 30$, $\kappa_+ = 15$, and the incident field is given by $u_{\text{inc}}(z) = e^{-i\kappa_+ \operatorname{Im}[z]}$.

2.5 Vekua theory

Unlike the Laplace case, we do not have rigorous theory for our Helmholtz solver, in spite of there being an abundance of empirical evidence that suggests similar convergence properties. This is analogous to the MFS for smooth problems, where convergence rates have been proven for a broad range of Laplace problems (see [97]), but we are not aware of a corresponding theorem for the Helmholtz case that establishes something similar to Conjecture 9 in [14]. In this section we will outline one potential strategy, which uses Vekua theory, for establishing a theoretical foundation for our Helmholtz solver, the further exploration of which will be deferred to future work. In Vekua theory the solution of an elliptic PDE in the plane is analytically continued in both real variables to the solution of a Goursat problem with characteristics prescribed by the original function in a new domain in $\mathbb{C}^2$. Loosely speaking, it is then observed that the Riemann–Volterra method provides a continuous mapping between the space of analytic functions and solutions to the original PDE. We will not provide a detailed

description of Vekua theory here, choosing instead to simply state the main relevant result. For a more comprehensive treatment of Vekua theory, we refer the reader to the book [157], the survey article [86], and the research articles [20, 53, 118, 122, 132].

The following is adapted from Theorem 1 in [146], which in turn is based on results in [157] and [53].

Theorem 2.7 (Still, 1992)**.** *Let $\Omega \subset \mathbb{C}$ be a bounded, simply connected open set with piecewise analytic boundary Γ with no degenerate corners, and let $z_* \in \Omega$ be fixed. Moreover, let $C^{k,\alpha}(\overline{\Omega})$ for $k \in \mathbb{N}$ and $\alpha \in (0,1]$ denote the space of functions on $\overline{\Omega}$ possessing continuous derivatives up to order k and whose k^{th}-partial derivatives are α-Hölder continuous. Then the following holds:*

1. *For every real-valued function $u \in C^2(\Omega) \cap C^{k,\alpha}(\overline{\Omega})$ for some $k \in \mathbb{N}$ and $\alpha \in (0,1]$ that satisfies $-\Delta u - \kappa^2 u = 0$ on Ω, there exists a unique function $\phi(z) \in C^{k,\alpha}(\overline{\Omega})$ analytic on Ω and $\phi(z_*) \in \mathbb{R}$ such that*

$$u(z) = \mathrm{Re}[\mathcal{V}(\phi; z)] \tag{2.71}$$

with

$$\mathcal{V}(\phi; z) = R(z, 0, z, \overline{z})\phi(z) - \int_{z_*}^{z} \phi(t) H(t, 0, z, \overline{z})\, dt \tag{2.72}$$

where R denotes the Riemann function corresponding to the Helmholtz equation:

$$R(t, \tau, z, \zeta) = J_0(\kappa\sqrt{(z-t)(\zeta-\tau)}) \text{ and } H(t, \tau, z, \zeta) = \frac{\partial}{\partial t} R(t, \tau, z, \zeta). \tag{2.73}$$

2. *For every function $\phi \in C^{k,\alpha}(\overline{\Omega})$ for some $k \in \mathbb{N}$ and $\alpha \in (0,1]$ analytic on Ω, the function*

$$u(z) = \mathrm{Re}[\mathcal{V}(\phi; z)] \tag{2.74}$$

is in $C^{k,\alpha}(\overline{\Omega})$ and satisfies $-\Delta u - \kappa^2 u = 0$ on Ω.

The mapping $\mathcal{V}$ in (2.72) is known as the Vekua transform and can be interpreted as a mapping from the space of analytic functions to solutions of the Helmholtz equation. This allows one to sometimes prove approximation results for solutions of PDEs using standard approximation theory. For example, using the identity

$$\mathcal{V}((z - z_*)^j; z) = j! \left(\frac{2}{\kappa}\right)^j J_j(\kappa|z - z_*|) e^{ij \arg(z - z_*)}, \tag{2.75}$$

a proof of which can be found in [146, Lemma 1], it can be shown that expansions of the form (2.44) will converge super-algebraically to smooth solutions of the Helmholtz

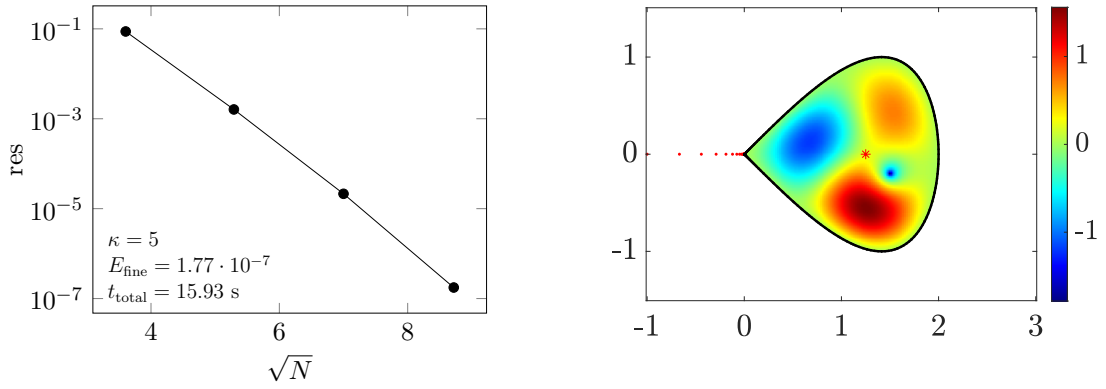


Figure 2.11: Convergence and total field plots for a sound-soft, interior problem in a teardrop shape

equation on a disk [53, Theorem 6.2]. This motivates studying an ansatz for the interior Helmholtz equation given by the Vekua transform of (2.16). Unfortunately, we cannot find a closed-form expression for the Vekua transform of a partial fraction term. In fact, numerical evidence suggests that this is not simply equal to a dipole. However, we can instead numerically approximate the Vekua transform. We illustrate this in Fig. 2.11 which shows the results when this procedure is applied to an interior Helmholtz problem on a teardrop-shaped domain with $\kappa = 5$ and boundary data generated by the incident field $u_{\text{inc}}(z) = Y_0(\kappa|z - 1.5 + 0.2i|)$, where the numerical integration is done using MATLAB's `integral` command. We empirically observe root-exponential convergence, raising the possibility of establishing a theory by clarifying the relationship between this ansatz and the one in (2.45).

Chapter 3

Direct solver for scattering in variable media[1]

3.1 Introduction

We now turn our attention to the case of scattering in inhomogeneous media in the plane. In particular, we consider a situation where an incident field u_{inc} impinges on a smooth (C^∞), compactly supported inhomogeneity in space described by a scattering potential

$$b(\boldsymbol{x}) = 1 - \left(\frac{v_{\text{free}}}{v(\boldsymbol{x})}\right)^2 \tag{3.1}$$

where v_{free} and v are the speeds of sound in free space and the perturbation, respectively. We are then interested in computing the scattered field, u. This situation is illustrated in Fig. 3.1.

The total field, $u_{\text{tot}} = u_{\text{inc}} + u$, for this situation satisfies the variable-coefficient Helmholtz equation

$$-\Delta u_{\text{tot}}(\boldsymbol{x}) - \kappa^2(1 - b(\boldsymbol{x}))u_{\text{tot}}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \mathbb{R}^2. \tag{3.2}$$

Using the fact that the incident field satisfies the constant-coefficient Helmholtz equation

$$-\Delta u_{\text{inc}}(\boldsymbol{x}) - \kappa^2 u_{\text{inc}}(\boldsymbol{x}) = 0 \tag{3.3}$$

and that the scattered field must satisfy the Sommerfeld radiation condition, then

[1]The material in this chapter was first presented in the preprint [65] by AG and his collaborator Per-Gunnar Martinsson (PGM). The idea of developing a solver for the Lippmann–Schwinger equation on uniform grids was suggested by PGM. AG and PGM jointly developed the solver, and AG ran all of the numerical experiments in this chapter.

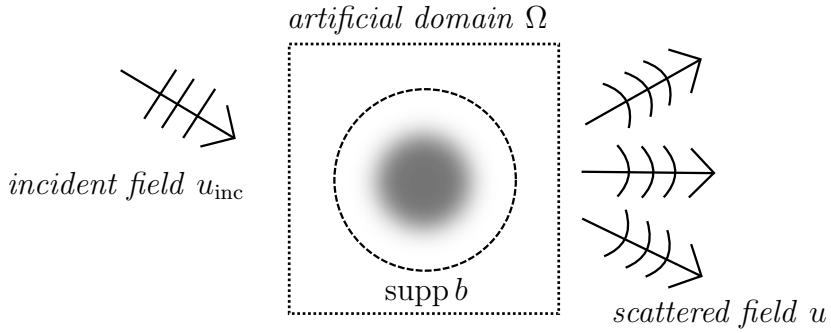


Figure 3.1: Setup for variable media scattering problem as considered in Chapter 3. The support of the scattering potential b is enclosed by the dashed circle, and the artificial domain Ω is the region inside of the dashed box.

yields the system of equations

$$\begin{cases} -\Delta u(\boldsymbol{x}) - \kappa^2(1 - b(\boldsymbol{x}))u(\boldsymbol{x}) = -\kappa^2 b(\boldsymbol{x})u_{\text{inc}}(\boldsymbol{x}), & \boldsymbol{x} \in \mathbb{R}^2, \\ \sqrt{r}\,(\partial_r u(\boldsymbol{x}) - i\kappa u(\boldsymbol{x})) = 0, & r = \|\boldsymbol{x}\| \to \infty. \end{cases} \tag{3.4}$$

There exist a few major difficulties in the numerical solution of (3.4) when using standard methods of directly discretizing the PDE, such as finite differences or finite elements. The most challenging in this environment is that the differential operator in the left-hand side of (3.4) trends increasingly ill-conditioned with the wavenumber, a reflection of the intrinsic ill-conditioning of the physics in the problem at high frequencies. Thus, to achieve even modest accuracy in the computed solution in the medium-to-high frequency regime, we must use high-order discretizations while keeping any spurious ill-conditioning introduced as an artifact of discretization to a minimum. Another challenge is that solutions to (3.4) become increasingly oscillatory as the wavenumber grows. As such, local discretizations suffer from dispersion errors, such as the pollution error in finite elements, which can necessitate a prohibitively large number of degrees of freedom [12]. Furthermore, (3.4) is posed on an unbounded domain, which necessitates special techniques, such as absorbing boundary conditions [54] or perfectly matched layers [18]. These can be nontrivial to implement with high accuracy in practice.

To deal with these challenges, it is advantageous to reformulate (3.4) as an integral equation. To this end, we introduce an artificial box Ω such that $\operatorname{supp} b \subset \Omega$ and use the ansatz

$$u(\boldsymbol{x}) = \int_\Omega G(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y})\, d\boldsymbol{y} \tag{3.5}$$

where G is the free-space fundamental solution of the Helmholtz equation in 2D given by

$$G(\boldsymbol{x}, \boldsymbol{y}) = \frac{i}{4} H_0(\kappa \|\boldsymbol{x} - \boldsymbol{y}\|) \tag{3.6}$$

where H_0 is the Hankel function of the first kind of order zero. Inserting (3.5) into (3.4) results in the *Lippmann–Schwinger equation*

$$\sigma(\boldsymbol{x}) + \kappa^2 b(\boldsymbol{x}) \int_\Omega G(\boldsymbol{x}, \boldsymbol{y}) \sigma(\boldsymbol{y})\, d\boldsymbol{y} = -\kappa^2 b(\boldsymbol{x}) u_{\text{inc}}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^2. \tag{3.7}$$

After the unknown density function σ is obtained through solving (3.7), the scattered field can be recovered using (3.5). If b is bounded above by 1, (3.7) is well-posed for all $\kappa > 0$ [44, Theorem 8.7]. Moreover, when the integral operator is posed on $L^2(\Omega)$, (3.7) is a Fredholm integral equation of the second kind [46], which makes it much easier to discretize without introducing spurious ill-conditioning than the differential operator. Also, the inclusion of the fundamental solution (3.6) in (3.5) prevents dispersion errors and automatically enforces the Sommerfeld radiation condition.

There are two major trade-offs when using such an integral equation formulation. The first is that the kernel (3.6) has a logarithmic singularity at the origin which requires special care in order to obtain high-order discretizations of (3.5) and (3.7). Fortunately, this issue has been the subject of prior work (e.g., [52, 158]). In our solver we will simply use an existing discretization scheme developed by Ran Duan and Vladimir Rokhlin in [52], based on local corrections to the trapezoidal rule. The other challenge is that (3.7) leads to a dense coefficient matrix upon discretization, as is typical with integral equations. The efficient inversion of this system will be the main subject of this chapter.

Perhaps the most common approach to numerically solving the systems that arise upon the discretization of integral equations is to combine a fast matrix-vector product algorithm, such as the Fast Multipole Method (FMM) [48, 72, 130] or the Fast Fourier Transform (FFT), with an iterative solver. This approach can be extremely effective for certain low-frequency problems since Fredholm integral equations of the second kind like (3.7) have eigenvalues that cluster at 1. However, for moderate-to-high values of κ or for highly refractive scattering potentials, the eigenvalues of the coefficient matrix are typically spread out in the complex plane. As a result, iterative solvers tend to converge slowly (cf. [52]). Furthermore, in some regimes it is of interest to compute scattered fields for multiple incident fields. This corresponds to solving linear systems with the same coefficient matrix and different right-hand sides, a regime in which iterative solvers typically do not derive much additional efficiency.

This motivates the study of direct solver techniques both for solution and preconditioning. The key observation that enables direct solvers in this setting is that while the coefficient matrices that arise are dense, they are highly rank structured. By this we mean that the matrices can be tessellated in such a way that certain blocks are approximately of low rank. This can then be exploited for efficient approximate inversion. One class of rank-structured matrices that is particularly well-suited for integral operators is the class of $\mathcal{H}$-matrices, which exploit low-rank structure in the off-diagonal blocks of the coefficient matrix [79, 81]. In the context of integral operators, this data sparsity is often attributable to the smoothness of the kernel away from the singularity and is closely related to ideas in the FMM. In this work we will use a special subclass of $\mathcal{H}$-matrices known in the literature as hierarchically block separable (HBS) or hierarchically semi-separable matrices. Our specific approach to exploiting rank structure can be traced back to a line of work originating in the papers [37,73,113,119,144] and related to the works [3,22], which applied similar ideas to the Lippmann–Schwinger equation in the contexts of adaptive mesh refinement and inverse scattering, respectively.

Loosely speaking, inversion algorithms for HBS matrices can be conceptually divided into two categories. The first class consists of methods that compute factors in a data-sparse inverse using explicit formulas in terms of the HBS factors. In this class are the many methods that rely on variants of the Woodbury formula (e.g., [4,28,62,101,113]) and explicit factorizations of the HBS matrix (e.g., [36,89,172]). These methods typically have very favorable constants but run the risk of encountering ill-conditioned matrices that need to be inverted in the algorithm. For example, the solver in [62] cannot reliably be used for our formulation of the Lippmann–Schwinger equation since it would require inverting singular matrices if the scattering potential were zero at any degree of freedom in our discretization. The other class of methods proceed by using the rank structure to form a larger but sparse linear system and then using a sparse direct solver to solve the resulting system (e.g., [5, 35, 47, 71, 88]). Through judicious choice of pivots, these methods can avoid inverting intermediary ill-conditioned matrices. However, there is a trade-off during pivot selection between stability and optimizing efficiency by minimizing fill-in. In our experience these method are incredibly robust but can be significantly slower and more memory intensive than methods from the previous class without careful tuning of the sparse solvers.

We introduce a new solver for the Lippmann–Schwinger equation, belonging to the first class, which is based on a discrete scattering matrix formulation similar to that presented in [37] and [28]. The solver utilizes uniform grids for efficient

compression and the specific geometry of the problem to accelerate inversion. Ample empirical evidence and heuristics suggest that the solver is as robust as the sparse matrix embedding approaches, while equally fast, if not faster, than most explicit inversion based approaches. The solver is especially efficacious when used to compute a coarse, approximate inverse for preconditioning. Moreover, the solver requires only compressing a discretization of the kernel, making it particularly well-suited for applications where multiple scattering potentials are of interest such as inverse scattering.

We now outline the remainder of this chapter. In Section 3.2 we describe the preliminaries necessary to describe our solver. Specifically, we recall some facts about Duan–Rokhlin quadratures, interpolative decompositions, iterative solvers for the Lippmann–Schwinger equation, and HBS matrices. Next is Section 3.3, where we describe how one can exploit the special structure of our discretization in order to efficiently compress the coefficient matrix into the HBS format. Section 3.4 describes our direct solver and includes complexity estimates and discussion about the relationship between our solver and a couple of other solvers from the literature. Then Section 3.5 explains the necessary considerations when using our solver as a preconditioner. Finally, Section 3.6 illustrates the efficacy of our solver in various regimes through numerical experiments.

Remark 3.1. In the literature the equation

$$u_{\text{tot}}(\boldsymbol{x}) + \kappa^2 \int_\Omega b(\boldsymbol{y}) G(\boldsymbol{x}, \boldsymbol{y}) u_{\text{tot}}(\boldsymbol{y})\, d\boldsymbol{y} = u_{\text{inc}}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^2, \tag{3.8}$$

which can be obtained by directly convolving (3.4) with (3.6), is also known as the Lippmann–Schwinger equation [44, Section 8.2]. In this formulation the scattered field can be solved for directly. However, in applications, gradients of the scattered field are often of interest, in which case the formulation (3.7) is slightly advantageous since the gradients of (3.5) can be computed without numerical differentiation. We note that (3.7) can actually be derived directly from (3.8) by scaling both sides of (3.8) by $-\kappa^2 b(\boldsymbol{x})$ and then substituting $\sigma(\boldsymbol{x}) = -\kappa^2 b(\boldsymbol{x}) u_{\text{tot}}(\boldsymbol{x})$. So, the formulations are clearly mathematically equivalent, and the solver developed in this chapter immediately extends to (3.8) as well.

3.2 Preliminaries

3.2.1 Interpolative decomposition

A particularly useful low-rank factorization in our application is the interpolative decomposition (ID), which uses selected columns (rows) of a matrix to approximately span its column (row) space. When the entries of the matrix can be computed on-the-fly, as in our case, this leads to considerable storage savings. In addition, the ID allows us to efficiently employ the method of proxy sources, described in Section 3.3. The ID is motivated by the following theorem, which is given as Theorem 1 in [113], which is in turn closely related to a result in [78]. A similar result can also be found in [38].

Theorem 3.1. *Let $\mathbf{A} \in \mathbb{C}^{m \times n}$ and let k be an integer such that $1 \leq k \leq \min(m, n)$. Then there exist matrices $\mathbf{V} \in \mathbb{C}^{n \times k}$, $\mathbf{E} \in \mathbb{C}^{m \times n}$, and an index vector $\mathcal{I} \subseteq [1, 2, \ldots, n]$ with $|\mathcal{I}| = k$ such that*

$$\mathbf{A} = \mathbf{A}(:, \mathcal{I})\mathbf{V}^* + \mathbf{E} \tag{3.9}$$

where

$$\|\mathbf{E}\| \leq \sigma_{k+1}(\mathbf{E})\sqrt{1 + k(n-k)}. \tag{3.10}$$

Furthermore, the matrix $\mathbf{V}$ can be taken to be of the form

$$\mathbf{V} = \mathbf{\Pi} \begin{bmatrix} \mathbf{I} \\ \mathbf{T} \end{bmatrix} \tag{3.11}$$

where $\mathbf{\Pi}$ is an $n \times n$ permutation matrix, $\mathbf{I}$ is the $k \times k$ identity, and $\mathbf{T} \in \mathbb{C}^{(n-k) \times k}$ satisfies the bound

$$\|\mathbf{T}\|_F \leq \sqrt{k(n-k)}. \tag{3.12}$$

For an index vector $\mathcal{I}$, we call a factorization

$$\mathbf{A} \approx \mathbf{A}(:, \mathcal{I})\mathbf{V}^* \tag{3.13}$$

a *column interpolative decomposition* and refer to the indices $\mathcal{I}$ as *skeleton indices*, the remainder $[1, 2, \ldots, n] \backslash \mathcal{I}$ as *residual indices*, and occasionally to $\mathbf{V}$ as the *projection matrix*. Theorem 3.1 then guarantees that for any matrix $\mathbf{A}$, there is a rank-k column ID with approximation error a multiplicative constant, depending only on k and the size of $\mathbf{A}$, from that of the optimal rank-k approximation as given by the Eckart-Young

theorem [92]. Furthermore, the condition (3.12) guarantees that the matrix $\mathbf{V}$ is well-conditioned. Similarly, a *row interpolative decomposition* is a factorization

$$\mathbf{A} \approx \mathbf{U}\mathbf{A}(\mathcal{I}, :), \tag{3.14}$$

where now $\mathbf{U}$ can be taken of the form of the right-hand side of (3.11), and we sometimes refer to it as the *interpolation matrix*. The relevant analog of Theorem 3.1 is straightforward.

It is not known if an ID satisfying (3.10) and (3.12) can be computed in polynomial time. Instead, we usually compute the ID using the column-pivoted QR (CPQR) factorization. Suppose a tall matrix $\mathbf{A}$ has a CPQR factorization of the form

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ & \mathbf{R}_{22} \end{bmatrix} \mathbf{\Pi}^* \tag{3.15}$$

where $\mathbf{Q} \in \mathbb{C}^{m \times n}$ has orthonormal columns, $\mathbf{\Pi}$ is an $n \times n$ permutation matrix, $\mathbf{R}_{11} \in \mathbb{C}^{k \times k}$ is upper-triangular, and $\mathbf{R}_{22} \in \mathbb{C}^{(n-k) \times (n-k)}$ and $\mathbf{R}_{12} \in \mathbb{C}^{k \times (n-k)}$ are dense. Then setting $\mathbf{Q}_1 = \mathbf{Q}(:, 1:k)$ and $\mathbf{Q}_2 = \mathbf{Q}(:, k+1:n)$, we have

$$\mathbf{A} = \mathbf{Q}_1 \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \end{bmatrix} \mathbf{\Pi}^* + \mathbf{Q}_2 \begin{bmatrix} \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \mathbf{\Pi}^* \tag{3.16}$$

$$= \mathbf{Q}_1 \mathbf{R}_{11} \begin{bmatrix} \mathbf{I} & \mathbf{R}_{11}^{-1} \mathbf{R}_{12} \end{bmatrix} \mathbf{\Pi}^* + \mathbf{Q}_2 \begin{bmatrix} \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \mathbf{\Pi}^*. \tag{3.17}$$

Defining

$$\mathcal{I} = \{i : \mathbf{\Pi}(i, j) = 1 \text{ for } 1 \leq j \leq k\} \text{ and } \mathbf{T} = \mathbf{R}_{11}^{-1} \mathbf{R}_{12}$$

allows us to rewrite (3.17) as

$$\mathbf{A} = \mathbf{A}(:, \mathcal{I}) \begin{bmatrix} \mathbf{I} & \mathbf{T} \end{bmatrix} \mathbf{\Pi}^* + \mathbf{Q}_2 \begin{bmatrix} \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \mathbf{\Pi}^*. \tag{3.18}$$

The first term in the right-hand side of (3.18) is clearly a rank-k ID. The approximation error depends only on the size of $\|\mathbf{R}_{22}\|$, which is precisely the approximation error incurred when a truncated CPQR factorization (3.15) is used as a low-rank approximation. While it is well known that there exist pathological examples when this error is much greater than the optimal rank-k approximation error (e.g., see Example 1 in [78], which is due to Kahan) such cases are extremely rare in practice, and we find in our applications IDs produced through this method yield factors that are only a small constant away from satisfying (3.10) and (3.12). Moreover, the dominant cost of computing the ID in this way is the cost of computing the CPQR factorization, which can be done using existing highly-optimized codes [7]. Since the CPQR algorithm can be terminated after k columns are orthogonalized, the total cost for computing the ID

this way is $\mathcal{O}(mnk)$. Using a randomized algorithm, this cost can be reduced almost to $\mathcal{O}(mn \log k + k^2 n)$ (see [166]). While this is obviously a better complexity in theory, we have observed that the parameters in this application are such that there is no significant benefit to switching to the randomized algorithm.

Remark 3.2. As discussed in [113], the algorithm in [78] can be used to compute an ID with factors that satisfy (3.9) when (3.10) is relaxed to

$$\|\mathbf{E}\| \leq \sigma_{k+1}(\mathbf{A})\sqrt{1 + nk(n-k)} \tag{3.19}$$

and (3.12) is relaxed to

$$\|\mathbf{T}\|_F \leq \sqrt{nk(n-k)}. \tag{3.20}$$

However, the CPQR-based algorithm discussed above is found to be sufficiently robust in practice and significantly more efficient than this algorithm.

3.2.2 High-order discretization

Without loss of generality, we assume that Ω is a square. Let $\{\boldsymbol{x}_j\}_{j=1}^N$ be a uniform grid on Ω with grid spacing h. We will use these points as both collocation points and quadrature points, corresponding to what is known in the literature as a Nyström discretization. Given such a grid and the fact that the scattering potential is both smooth and compactly supported, the most straightforward method of discretizing (3.7) is to use a "punctured trapezoidal rule", which would yield a collocation condition of the form

$$\sigma_i + \kappa^2 h^2 b(\boldsymbol{x}_i) \sum_{\substack{j=1 \\ j \neq i}}^{N} G(\boldsymbol{x}_i, \boldsymbol{x}_j)\sigma_j = -\kappa^2 b(\boldsymbol{x}_i) u_{\text{inc}}(\boldsymbol{x}_i) \tag{3.21}$$

where we expect $\sigma_i \approx \sigma(\boldsymbol{x}_i)$ for $i = 1, 2, \ldots, N$ and sufficiently large N. The $i = j$ index is omitted in the summand in (3.21) due to the singularity of the kernel, resulting in a scheme that converges at a rate of $\mathcal{O}(h^2 \log(1/h)^2)$ [52]. Fortunately, it was shown in [52] that the convergence order can be improved by simply adding some local corrections. This results in the modified collocation condition

$$\sigma_i + \kappa^2 h^2 b(\boldsymbol{x}_i) \left[\sum_{\substack{j=1 \\ j \neq i}}^{N} G(\boldsymbol{x}_i, \boldsymbol{x}_j)\sigma_j + \sum_{j \in \mathcal{S}_i} \tau_{ij}\sigma_j \right] = -\kappa^2 b(\boldsymbol{x}_i) u_{\text{inc}}(\boldsymbol{x}_i) \tag{3.22}$$

where $\mathcal{S}_i$ is a set that contains the indices of some points in a local stencil around $\boldsymbol{x}_i$ for $i = 1, 2, \ldots, N$. The number of elements in each $\mathcal{S}_i$ depends only on the order of the

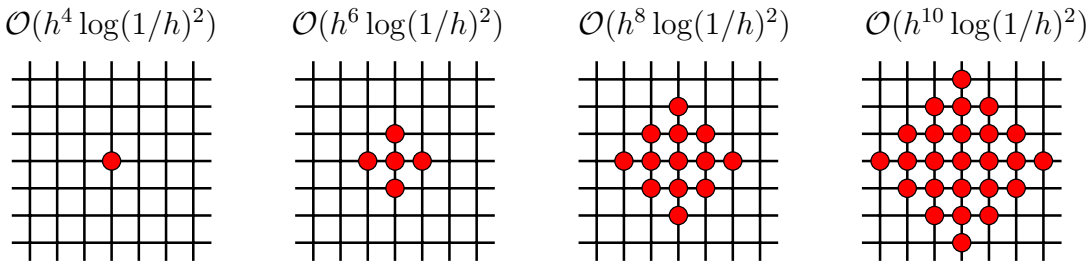


Figure 3.2: The stencils of the Duan–Rokhlin quadrature corrections described in Section 3.2.2. The index set $\mathcal{S}_i$ is shown with red dots on a local grid centered around $\boldsymbol{x}_i$. The 4^{th}-order stencil (leftmost) requires only a single correction point, while the 6^{th}-, 8^{th}-, and 10^{th}-order stencils require 5, 13, and 25 points, respectively.

quadrature rule, and it was found in [52] that an $\mathcal{O}(h^4 \log(1/h)^2)$-rule can be obtained with just $\mathcal{S}_i = \{i\}$ (i.e., a diagonal correction). Fig. 3.2 illustrates the local correction stencils for higher-order corrections. In the remainder we will informally refer to the order of the rule by just its polynomial part (e.g., we refer to the $\mathcal{O}(h^4 \log(1/h)^2)$-rule as a 4^{th}-order rule).

Clearly, the condition in (3.22) can be expressed as a linear system

$$(\mathbf{I} + \mathbf{BG} + \mathbf{BT})\mathbf{q} = \mathbf{f} \tag{3.23}$$

where $\mathbf{I}$ is the $N \times N$ identity, $\mathbf{B}$ is diagonal with entries given by $\mathbf{B}(i,j) = \delta_{ij}\kappa^2 b(\boldsymbol{x}_i)$, $\mathbf{q}(i) = \sigma_i$, $\mathbf{f}(i) = -\kappa^2 b(\boldsymbol{x}_i) u_{\text{inc}}(\boldsymbol{x}_i)$, and $\mathbf{G}$ and $\mathbf{T}$ are matrices with entries given by

$$\mathbf{G}(i,j) = \begin{cases} 0, & i = j, \\ h^2 G(\boldsymbol{x}_i, \boldsymbol{x}_j), & i \neq j, \end{cases} \text{ and } \mathbf{T}(i,j) = \begin{cases} h^2 \tau_{ij}, & j \in \mathcal{S}_i \\ 0, & j \notin \mathcal{S}_i \end{cases} \tag{3.24}$$

for $i, j = 1, 2, \ldots, N$.

3.2.3 Iterative solution

Since we have used a uniform grid for our discretization and the kernel (3.6) and quadrature corrections in (3.22) are translation invariant, the matrices $\mathbf{G}$ and $\mathbf{T}$ in (3.23) are Toeplitz. Therefore, the coefficient matrix in (3.23) is the sum of the identity matrix and a diagonal matrix times a Toeplitz matrix, and can be applied to a vector in quasilinear complexity using the FFT. This observation was coupled with an iterative solver in [52] and shown to be extraordinarily effective in certain regimes. To illustrate this, we consider the coefficient matrices of two model problems, the first being a case where the scattering potential is a Gaussian bump (see Fig. 3.3(a)) given by

$$b(x_1, x_2) = -1.5 \exp\left(-80\sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}\right), \tag{3.25}$$

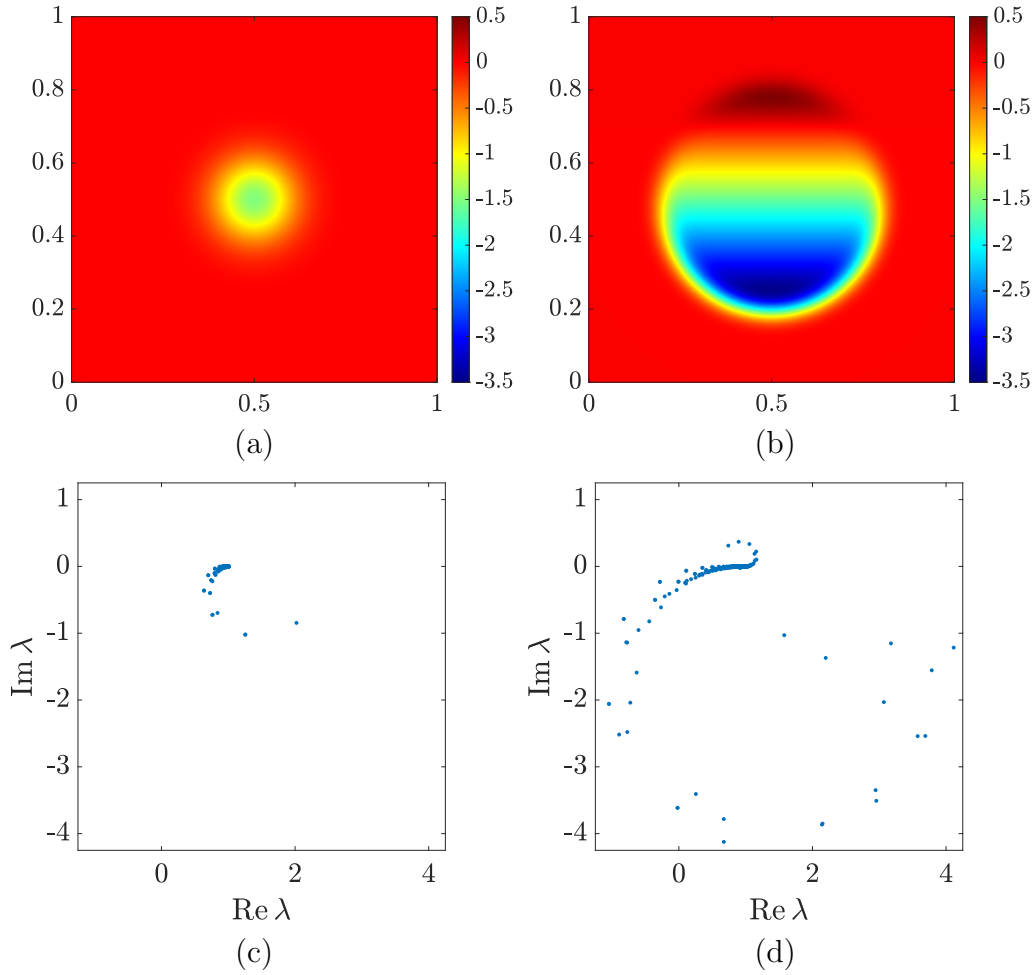


Figure 3.3: (a) Gaussian bump scattering potential. (b) Vertically-graded lens scattering potential. (c) Spectra of coefficient matrix for the Gaussian bump. (d) Spectra of coefficient matrix for the lens

and the other being a case where the scattering potential is a vertically-graded lens (see Fig. 3.3(b)) from [61] given by

$$b(x_1, x_2) = 4(x_2 - 0.7)\left(1 - \operatorname{erf}\left(25\left(\sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} - 0.3\right)\right)\right). \quad (3.26)$$

We consider a problem in the low-frequency regime where $\kappa = 8\pi$ (so that Ω is 4λ across) and use a discretization with $N = 1600$. The spectra of the coefficient matrices with the 10^{th}-order Duan–Rokhlin corrections is shown in Fig. 3.3(c-d). We see that in both cases the spectra accumulate at 1, reflecting the fact that (3.7) is a Fredholm equation of the second kind. For the Gaussian bump we see that all of the eigenvalues are clustered, and so assuming that the eigenvectors are reasonably well-conditioned, we expect Krylov solvers to converge quickly for this problem. Indeed, GMRES

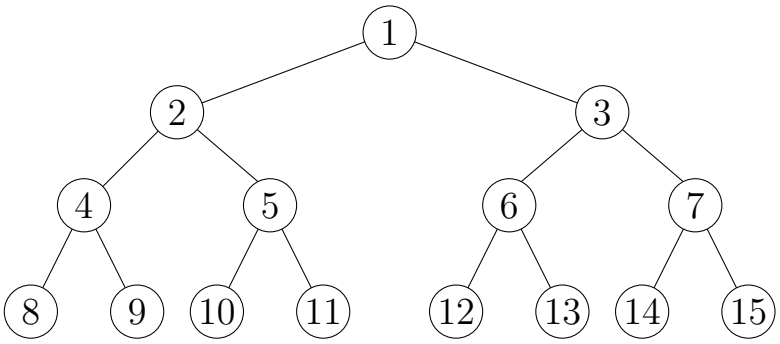


Figure 3.4: A uniform binary tree with four levels

converges to a tolerance of 10^{-10} in only 15 iterations in this case. However, we see that the eigenvalues for the lens problem are much more spread out in the plane, and now GMRES takes 79 iterations to converge to the same tolerance.

This is an example of the more general phenomenon that iterative solvers tend to converge slowly for scattering problems that exhibit a substantial amount of backscattering. Naturally, this problem is exacerbated as the wavenumber increases since (3.7) then tends to a Fredholm equation of the first kind. This motivates the need for preconditioners or direct solvers.

3.2.4 Hierarchically block separable matrices

To exploit rank structure, we need a hierarchical partitioning of the coefficient matrix, which we will specify by a binary tree on the indices. For an $N \times N$ matrix $\mathbf{A}$, we associate the index vector $\mathcal{I}_1 = [1, 2, \ldots, N]$ with the *root* node of the tree, which we label by 1. We then subdivide $\mathcal{I}_1$ into two disjoint index vectors $\mathcal{I}_2$ and $\mathcal{I}_3$ (i.e., $\mathcal{I}_1 = \mathcal{I}_2 \cup \mathcal{I}_3$ and $\mathcal{I}_2 \cap \mathcal{I}_3 = \emptyset$), which we associate with the two *children* nodes of node 1 and label as 2 and 3, respectively. We further subdivide $\mathcal{I}_2$ into disjoint index vectors $\mathcal{I}_4$ and $\mathcal{I}_5$ to get the children nodes 4 and 5 of the node 2; we obtain the children of node 3, which we denote nodes 6 and 7, by the analogous division of $\mathcal{I}_3$. This process of subdivision is repeated until the index vectors corresponding to the nodes with no children, which we call *leaves*, contain sufficiently few points. We use a convention where the root is on level 0, the children of the root are on level 1, and so on, so that the leaves are on level L. Such a tree with 4 levels (i.e., $L = 3$) is illustrated in Fig. 3.4. For our purposes it will suffice to use uniform binary trees. That is, there will be exactly 2^ℓ nodes on level ℓ for $\ell = 0, 1, \ldots, L$.

Given such a tree, we can impose a rank-structured format.

Definition 3.2. We say that $\mathbf{A} \in \mathbb{C}^{N\times N}$ is a *hierarchically block separable (HBS) matrix* given a binary tree structure on the indices of the aforementioned form provided that the following two conditions are met:

1. For any two sibling nodes α and β in the tree, there exist matrices $\widehat{\mathbf{U}}_\alpha \in \mathbb{C}^{n_\alpha\times k_\alpha}$, $\mathbf{A}_{\alpha,\beta} \in \mathbb{C}^{k_\alpha\times k_\beta}$, and $\widehat{\mathbf{V}}_\beta \in \mathbb{C}^{n_\beta\times k_\beta}$ such that

$$\mathbf{A}(\mathcal{I}_\alpha, \mathcal{I}_\beta) = \widehat{\mathbf{U}}_\alpha \mathbf{A}_{\alpha,\beta} \widehat{\mathbf{V}}_\beta^* \tag{3.27}$$

   where $k_\alpha < n_\alpha = |\mathcal{I}_\alpha|$ and $k_\beta < n_\beta = |\mathcal{I}_\beta|$.

2. For any non-leaf node τ in the tree with children nodes α and β, there exist matrices $\mathbf{U}_\tau \in \mathbb{C}^{(k_\alpha+k_\beta)\times k_\tau}$ and $\mathbf{V}_\tau \in \mathbb{C}^{(k_\alpha+k_\beta)\times k_\tau}$, such that

$$\widehat{\mathbf{U}}_\tau = \begin{bmatrix} \widehat{\mathbf{U}}_\alpha & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{U}}_\beta \end{bmatrix} \mathbf{U}_\tau \text{ and } \widehat{\mathbf{V}}_\tau = \begin{bmatrix} \widehat{\mathbf{V}}_\alpha & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{V}}_\beta \end{bmatrix} \mathbf{V}_\tau. \tag{3.28}$$

   For notational convenience we also define $\mathbf{A}_\tau = \mathbf{A}(\mathcal{I}_\tau, \mathcal{I}_\tau)$ for every node τ, and $\mathbf{U}_\tau = \widehat{\mathbf{U}}_\tau$ and $\mathbf{V}_\tau = \widehat{\mathbf{V}}_\tau$ for every leaf node τ.

The first condition in Definition 3.2 enforces that $\mathbf{A}$ can be tessellated in such a fashion that certain off-diagonal blocks of the matrix $\mathbf{A}$ are of low rank. This by itself is the definition of the hierarchically off-diagonal low-rank (HODLR) format [4]. The second condition requires that the basis matrices are hierarchically nested in the sense of (3.28), which distinguishes the HBS format from the HODLR format.

To illustrate Definition 3.2, we consider the example of a 3-level binary tree with the labeling convention in Fig. 3.4. Considering a block partitioning on level 1 in the tree gives the relation

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_2 & \mathbf{A}(\mathcal{I}_2, \mathcal{I}_3) \\ \mathbf{A}(\mathcal{I}_3, \mathcal{I}_2) & \mathbf{A}_3 \end{bmatrix} \tag{3.29}$$

$$= \begin{bmatrix} \mathbf{A}_2 & \\ & \mathbf{A}_3 \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{U}}_2 & \\ & \widehat{\mathbf{U}}_3 \end{bmatrix} \begin{bmatrix} & \mathbf{A}_{2,3} \\ \mathbf{A}_{3,2} & \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{V}}_2^* & \\ & \widehat{\mathbf{V}}_3^* \end{bmatrix} \tag{3.30}$$

$$= \begin{bmatrix} \mathbf{A}_2 & \\ & \mathbf{A}_3 \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{U}}_4 & & & \\ & \widehat{\mathbf{U}}_5 & & \\ & & \widehat{\mathbf{U}}_6 & \\ & & & \widehat{\mathbf{U}}_7 \end{bmatrix} \begin{bmatrix} \mathbf{U}_2 & \\ & \mathbf{U}_3 \end{bmatrix} \begin{bmatrix} & \mathbf{A}_{2,3} \\ \mathbf{A}_{3,2} & \end{bmatrix} \begin{bmatrix} \mathbf{V}_2^* & \\ & \mathbf{V}_3^* \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{V}}_4^* & & & \\ & \widehat{\mathbf{V}}_5^* & & \\ & & \widehat{\mathbf{V}}_6^* & \\ & & & \widehat{\mathbf{V}}_7^* \end{bmatrix}. \tag{3.31}$$

In (3.30) and (3.31) we have used conditions 1 and 2 in Definition 3.2, respectively. The hierarchical nature of the tree then allows us to further subdivide the first term

in (3.31) if we move our consideration to level 2 of the tree. This allows the further factorization

$$\begin{bmatrix} \mathbf{A}_2 & \\ & \mathbf{A}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_4 & \mathbf{A}(\mathcal{I}_4, \mathcal{I}_5) & & \\ \mathbf{A}(\mathcal{I}_5, \mathcal{I}_4) & \mathbf{A}_5 & & \\ & & \mathbf{A}_6 & \mathbf{A}(\mathcal{I}_6, \mathcal{I}_7) \\ & & \mathbf{A}(\mathcal{I}_7, \mathcal{I}_6) & \mathbf{A}_7 \end{bmatrix} \tag{3.32}$$

$$= \begin{bmatrix} \mathbf{A}_4 & & & \\ & \mathbf{A}_5 & & \\ & & \mathbf{A}_6 & \\ & & & \mathbf{A}_7 \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{U}}_4 & & & \\ & \widehat{\mathbf{U}}_5 & & \\ & & \widehat{\mathbf{U}}_6 & \\ & & & \widehat{\mathbf{U}}_7 \end{bmatrix} \begin{bmatrix} & \mathbf{A}_{4,5} & & \\ \mathbf{A}_{5,4} & & & \\ & & & \mathbf{A}_{6,7} \\ & & \mathbf{A}_{7,6} & \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{V}}_4^* & & & \\ & \widehat{\mathbf{V}}_5^* & & \\ & & \widehat{\mathbf{V}}_6^* & \\ & & & \widehat{\mathbf{V}}_7^* \end{bmatrix}. \tag{3.33}$$

Inserting (3.33) into (3.31) then yields the following "telescoping" factorization

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_4 & & & \\ & \mathbf{A}_5 & & \\ & & \mathbf{A}_6 & \\ & & & \mathbf{A}_7 \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{U}}_4 & & & \\ & \widehat{\mathbf{U}}_5 & & \\ & & \widehat{\mathbf{U}}_6 & \\ & & & \widehat{\mathbf{U}}_7 \end{bmatrix} \left(\begin{bmatrix} & \mathbf{A}_{4,5} & & \\ \mathbf{A}_{5,4} & & & \\ & & & \mathbf{A}_{6,7} \\ & & \mathbf{A}_{7,6} & \end{bmatrix} + \begin{bmatrix} \mathbf{U}_2 & \\ & \mathbf{U}_3 \end{bmatrix} \begin{bmatrix} & \mathbf{A}_{2,3} \\ \mathbf{A}_{3,2} & \end{bmatrix} \begin{bmatrix} \mathbf{V}_2^* & \\ & \mathbf{V}_3^* \end{bmatrix} \right) \begin{bmatrix} \widehat{\mathbf{V}}_4^* & & & \\ & \widehat{\mathbf{V}}_5^* & & \\ & & \widehat{\mathbf{V}}_6^* & \\ & & & \widehat{\mathbf{V}}_7^* \end{bmatrix}. \tag{3.34}$$

If the tree were to have more levels, then the first term in the right-hand side of (3.34) would have further rank deficiencies e.g.,

$$\mathbf{A}_4 = \begin{bmatrix} \mathbf{A}_8 & \mathbf{A}(\mathcal{I}_8, \mathcal{I}_9) \\ \mathbf{A}(\mathcal{I}_9, \mathcal{I}_8) & \mathbf{A}_9 \end{bmatrix} \tag{3.35}$$

$$= \begin{bmatrix} \mathbf{A}_8 & \\ & \mathbf{A}_9 \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{U}}_8 & \\ & \widehat{\mathbf{U}}_9 \end{bmatrix} \begin{bmatrix} & \mathbf{A}_{8,9} \\ \mathbf{A}_{9,8} & \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{V}}_8^* & \\ & \widehat{\mathbf{V}}_9^* \end{bmatrix}, \tag{3.36}$$

and the basis matrices for the nodes 4, 5, 6, and 7 would be further nested e.g.,

$$\widehat{\mathbf{U}}_4 = \begin{bmatrix} \widehat{\mathbf{U}}_8 & \\ & \widehat{\mathbf{U}}_9 \end{bmatrix} \mathbf{U}_4. \tag{3.37}$$

It is clear from (3.34), (3.36), and (3.37) that HBS matrices can be factorized into all block diagonal factors and a matrix on the root level of the tree of the form

$$\begin{bmatrix} & \mathbf{A}_{2,3} \\ \mathbf{A}_{3,2} & \end{bmatrix}. \tag{3.38}$$

For a more detailed treatment of HBS matrices, we refer the reader to the survey paper [62] and the textbook [112].

Remark 3.3. In the first condition in Definition 3.2, we have required the off-diagonal blocks to be mathematically of low rank without any constraints on how small the ranks

should be. In practice k_α and k_β are numerical ranks which should be considerably smaller than n_α and n_β, respectively. From the factorization (3.34), it is clear that smaller ranks correspond to more data sparsity, and so it should be unsurprising that the complexity of HBS matrix algebra depends on the ranks.

Remark 3.4. From Definition 3.2 it follows that the only matrices necessary to specify an HBS matrix are:

type of node	matrix	dimensions
leaf node τ	$\mathbf{A}_\tau$	$n_\tau \times n_\tau$
	$\mathbf{U}_\tau$	$n_\tau \times k_\tau$
	$\mathbf{V}_\tau$	$n_\tau \times k_\tau$
non-root parent node τ	$\mathbf{A}_{\alpha,\beta}$	$k_\alpha \times k_\beta$
with children α and β	$\mathbf{A}_{\beta,\alpha}$	$k_\beta \times k_\alpha$
	$\mathbf{U}_\tau$	$(k_\alpha + k_\beta) \times k_\tau$
	$\mathbf{V}_\tau$	$(k_\alpha + k_\beta) \times k_\tau$
root node τ with children α and β	$\mathbf{A}_{\alpha,\beta}$	$k_\alpha \times k_\beta$
	$\mathbf{A}_{\beta,\alpha}$	$k_\beta \times k_\alpha$

Note that the $n_\tau \times k_\tau$ large "extended" basis matrices $\widehat{\mathbf{U}}_\tau$ and $\widehat{\mathbf{V}}_\tau$ do not need to be formed explicitly for any non-leaf node τ, which is essential for efficiency since these matrices have $\mathcal{O}(N)$ rows on the upper levels of the tree.

Remark 3.5. Both the HODLR and HBS formats can be placed into the broader framework of $\mathcal{H}$-matrices introduced by Hackbusch and collaborators [69,79,82] and are closely related to earlier work by Greengard and Rokhlin on the FMM [72]. $\mathcal{H}$-matrix formats can be divided into two categories, strongly-admissible formulations, which treat off-diagonal blocks "near" the diagonal as dense matrices, and weakly-admissible formulations, which do not. $\mathcal{H}$-matrices with a nested basis structure are referred to as $\mathcal{H}^2$-matrices [23–25, 81]. Therefore, HODLR and HBS matrices are weakly-admissible $\mathcal{H}$- and $\mathcal{H}^2$-matrices, respectively. While weakly-admissibile formats generally lead to higher ranks in the off-diagonal blocks, they allow for simplified inversion formulas with smaller constants, a trade-off we empirically find decisively advantageous for 2D problems. Similarly, the use of the HBS format over the HODLR format also increases the ranks but often allows for the removal of $\log N$ terms in the storage and arithmetic complexities [13]. For a more thorough treatment of $\mathcal{H}$-matrices, we refer the reader to the textbooks [17, 80] and the survey article [13].

3.3 HBS compression

To compress the coefficient matrix, we compute an HBS approximation to just the matrix $\mathbf{G} + \mathbf{T}$ in (3.23). This has the advantage of making the *compression stage* of the solver independent of the scattering potential, which is useful in some applications such as inverse scattering and allows us to leverage translation invariance to accelerate the procedure. For now we restrict our discussion to the case of compressing just $\mathbf{G}$ and defer discussion of the additional considerations needed when the quadrature corrections are included to the end of this section. Thus, given a compression tolerance ε, we seek an HBS matrix $\mathbf{G}^{\varepsilon}$ such that

$$\frac{\|\mathbf{G} - \mathbf{G}^{\varepsilon}\|}{\|\mathbf{G}\|} \leq \varepsilon. \tag{3.39}$$

From the entries in (3.24), we see that $\mathbf{G}$ can be interpreted as encoding an N-body problem. Particularly, a matrix-vector product with $\mathbf{G}$ and a vector $\mathbf{q}$ can be interpreted as computing the field at the target points $\{\boldsymbol{x}_j\}_{j=1}^N$ generated by sources at these same points with charges prescribed by the vector $\mathbf{q}$ scaled by h^2. The kernel specifying the interactions is just the free-space fundamental solution of the Helmholtz equation (3.6). Since (3.6) is smooth when $\boldsymbol{x} \neq \boldsymbol{y}$, it is unsurprising that there is a redundancy in information when we limit our consideration to given sets of sources $\{\boldsymbol{x}_j\}_{j\in\mathcal{I}_\mathrm{s}}$ and targets $\{\boldsymbol{x}_j\}_{j\in\mathcal{I}_\mathrm{t}}$ that are separated in space. In terms of linear algebra, this means that the we should expect the submatrices $\mathbf{G}(\mathcal{I}_\mathrm{t}, \mathcal{I}_\mathrm{s})$ and $\mathbf{G}(\mathcal{I}_\mathrm{s}, \mathcal{I}_\mathrm{t})$ to be of low numerical rank. Leveraging this redundancy is one of the key ideas in the FMM, which uses multipole and harmonic expansions to efficiently represent the interactions between particles in well-separated boxes. Instead of using an expansion based in special functions, we will use custom expansions produced by low-rank factorizations. This results in greater compression and allows us to use a weakly-admissible format, such as the HBS format.

As discussed in Section 3.2.4, this requires a tessellation of the matrix where various off-diagonal blocks are of low rank. For the coefficient matrix in (3.23), this can be obtained by basing the binary tree required in Definition 3.2 on a division of physical space. As always, we let the index vector associated with the root node contain all of the indices, and so $\mathcal{I}_1 = [1, 2, \ldots, N]$. To obtain the index vectors associated with the children of the root node, we divide Ω in half using a vertical cut and let Ω_2 and Ω_3 denote the left and right boxes, respectively. $\mathcal{I}_2$ and $\mathcal{I}_3$ contain the indices of points lying in Ω_2 and Ω_3, respectively, e.g.,

$$\mathcal{I}_2 = \{j \in [1, 2, \ldots, N] : \boldsymbol{x}_j \in \Omega_2\}. \tag{3.40}$$

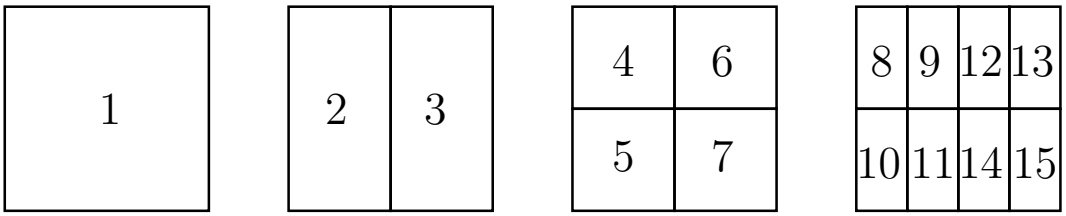


Figure 3.5: Division of Ω into boxes corresponding to the tree in Fig. 3.4

We proceed by making a horizontal cut on Ω_2 to produce boxes Ω_4 and Ω_5 on the top and bottom and populate $\mathcal{I}_4$ and $\mathcal{I}_5$ with the indices of points in Ω_4 and Ω_5, respectively. The analogous procedure of horizontally dividing the box Ω_3 produces Ω_6, Ω_7, $\mathcal{I}_6$, and $\mathcal{I}_7$. We continue this process, dividing all of the smallest boxes at each step and alternating between vertical and horizontal divisions, until each of the boxes on the leaf level contains fewer than N_{dense} points where N_{dense} is sufficiently small that dense, cubic linear algebra can be done rapidly on $N_{\text{dense}} \times N_{\text{dense}}$ matrices. The division of Ω into boxes for a four level tree such as Fig. 3.4 is illustrated in Fig. 3.5.

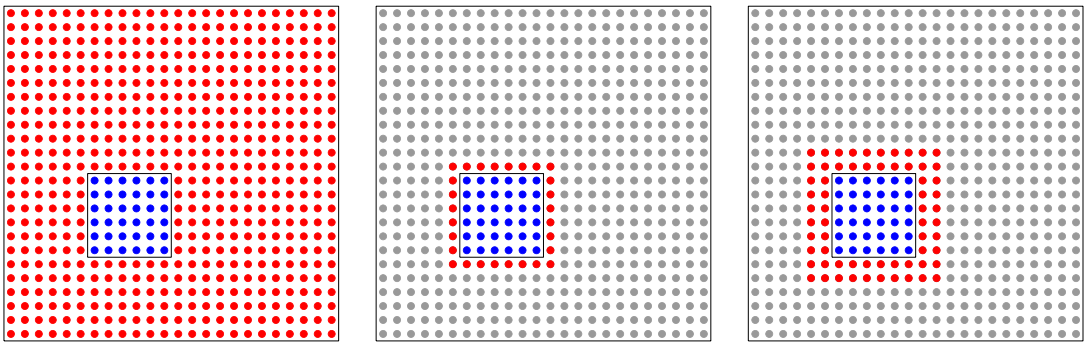
The computation of an HBS approximation to $\mathbf{G}$ will be done using a single upward pass through the tree, and we will use the ID for reasons stated in Section 3.2.1. Starting with a leaf node τ, we seek to compute skeleton indices $\mathcal{I}_\tau^{\text{skel}}$ and a basis matrix $\mathbf{U}_\tau$ such that

$$\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c) \approx \mathbf{U}_\tau \mathbf{G}(\mathcal{I}_\tau^{\text{skel}}, \mathcal{I}_\tau^c). \tag{3.41}$$

In (3.41) we must compute a factorization of a $|\mathcal{I}_\tau| \times |\mathcal{I}_\tau^c|$ matrix. Since $|\mathcal{I}_\tau^c| = \mathcal{O}(N)$, this is a rather expensive operation. Fortunately, the computation of $\mathcal{I}_\tau^{\text{skel}}$ and $\mathbf{U}_\tau$ can be accelerated using the *method of proxy sources* [38]. The key observation here is that in the above physical interpretation the matrix $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c)$ encodes the interactions between sources on the points $\{\boldsymbol{x}_j\}_{j \in \mathcal{I}_\tau^c}$ and targets on the points $\{\boldsymbol{x}_j\}_{j \in \mathcal{I}_\tau}$. However, potential theory states that all such interactions can be replicated by a layer potential placed on the boundary of the box Ω_τ [112, Sec. 17.2], which can then be discretized using a ring of nearby collocation points indexed by $\mathcal{I}_\tau^{\text{proxy}}$. This is illustrated in Fig. 3.6(b), where the points in the box Ω_τ are denoted in blue and the points $\{\boldsymbol{x}_j\}_{j \in \mathcal{I}_\tau^{\text{proxy}}}$ in red. In linear algebra terms, the claim is that the columns of $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c)$ and $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^{\text{proxy}})$ span approximately the same space. As such, if we can compute $\mathbf{U}_\tau$ and $\mathcal{I}_\tau^{\text{skel}}$ such that

$$\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^{\text{proxy}}) \approx \mathbf{U}_\tau \mathbf{G}(\mathcal{I}_\tau^{\text{skel}}, \mathcal{I}_\tau^{\text{proxy}}) \tag{3.42}$$

to some tolerance, we expect the same $\mathbf{U}_\tau$ and $\mathcal{I}_\tau^{\text{skel}}$ to also satisfy (3.41).



(a) (b) (c)

Figure 3.6: Illustration of the method of proxy sources when we seek to build a basis for the range of the matrix $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c)$, which maps sources on the red points (indexed by $\mathcal{I}_\tau^c$) to targets on the blue points (indexed by $\mathcal{I}_\tau$) in (a). Since $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c)$ has $\mathcal{O}(N)$ columns, this is prohibitively expensive. Instead, we may consider the matrix $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^{\text{proxy}})$ with $\mathcal{I}_\tau^{\text{proxy}}$ indexing the red points in (b). An argument using Green's theorem states that the column space of $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^{\text{proxy}})$ approximates that of $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c)$. If $\mathcal{I}_\tau^{\text{proxy}}$ were instead chosen to be the red points in (c), higher accuracy can be obtained.

	Size of Ω_τ		
	$0.25\lambda \times 0.25\lambda$	$1\lambda \times 1\lambda$	$4\lambda \times 4\lambda$
width=1	$8.7 \cdot 10^{-5}$	$1.6 \cdot 10^{-4}$	$9.6 \cdot 10^{-4}$
width=2	$1.1 \cdot 10^{-10}$	$1.8 \cdot 10^{-10}$	$6.2 \cdot 10^{-10}$
width=3	$5.6 \cdot 10^{-15}$	$5.2 \cdot 10^{-15}$	$4.0 \cdot 10^{-15}$

Table 3.1: Errors incurred when using the method of proxy sources for a box with 20×20 points. The error reported is $\|\mathbf{H} - \mathbf{Q}\mathbf{Q}^*\mathbf{H}\|_{\text{C}} / \|\mathbf{H}\|_{\text{C}}$ where $\mathbf{H} = \mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c)$ is the matrix to be approximated where $\mathbf{Q}$ is a matrix whose columns form an orthornormal basis for the columns of $\mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^{\text{proxy}})$ and $\|\mathbf{H}\|_{\text{C}} = \max |\mathbf{H}(i,j)|$.

The error incurred in using the method of proxy sources depends on how accurately we have discretized the continuous layer potential. In our geometry this relates directly to the "thickness" of the proxy ring. If the ring is chosen to be one point thick as in Fig. 3.6(b), we typically achieve 4–5 digits of accuracy. A ring that is two points thick as in Fig. 3.6(c) typically achieves 9–10 digits and full double precision is typically achieved for a ring that is three or more points thick. This is illustrated in Table 3.1 for a target box containing 20×20 points with three different wavenumbers.

A significant further acceleration can be obtained if we choose the number of total degrees of freedom to be divisible by the number of leaf nodes which is what we do in practice. Since the kernel and quadrature corrections are translation invariant, the matrix that must be factorized for each leaf box when the proxy ring is used is the

same. Moreover, since $\mathbf{G}$ is symmetric (but not Hermitian), we can take $\mathbf{V}_\tau = \mathbf{U}_\tau$ and use the skeleton indices $\mathcal{I}_\tau^{\text{skel}}$ for the column skeletons as well. We can therefore compute (3.42) for just a single leaf in order to obtain all the necessary factors for nodes at level L.

Next, we compute the necessary factors for the nodes on level $L-1$. Let τ be a node on this level with leaf children α and β. The analog of (3.41) is then

$$\mathbf{G}([\mathcal{I}_\alpha^{\text{skel}}, \mathcal{I}_\beta^{\text{skel}}], \mathcal{I}_\tau^c) \approx \mathbf{U}_\tau \mathbf{G}(\mathcal{I}_\tau^{\text{skel}}, \mathcal{I}_\tau^c). \tag{3.43}$$

The sibling-sibling interaction matrices are then given by

$$\mathbf{G}_{\alpha,\beta} = \mathbf{G}(\mathcal{I}_\alpha^{\text{skel}}, \mathcal{I}_\beta^{\text{skel}}) \quad \text{and} \quad \mathbf{G}_{\beta,\alpha} = \mathbf{G}(\mathcal{I}_\beta^{\text{skel}}, \mathcal{I}_\alpha^{\text{skel}}). \tag{3.44}$$

The method of proxy sources, translation invariance, and symmetry can again be leveraged so that it suffices to compute just a single ID on this level. We proceed by moving up the tree, each time computing a single factorization analogous to (3.43). When the root level of the tree is reached, all factors needed to specify the HBS matrix, as tabulated in Remark 3.4, will have been computed.

Since $\mathbf{T}$ is diagonal for the 4^{th}-order Duan–Rokhlin quadrature corrections and only off-diagonal blocks are necessary to compute the HBS factors, including these corrections leaves the process completely unchanged. Higher-order corrections do, however, affect the off-diagonal blocks. Since these corrections are local, they do not affect the asymptotic scaling of the ranks, but they can substantially increase the constants. This is illustrated in Fig. 3.7 where the ranks and skeleton points when the interactions on a box with 20×20 points with $\kappa = 10$ are compressed using the ID when the 4^{th}- and 10^{th}-order quadrature corrections are used. We see that for both compression tolerances, the ranks for the 10^{th}-order corrections are roughly 60% higher than that for the 4^{th}-order corrections. The only other effect of including quadrature corrections is an increase in the thickness of the proxy ring required to obtain a fixed accuracy. For example, the 10^{th}-order quadrature corrections require a proxy ring that is 4 points thick to achieve machine accuracy in double precision.

Remark 3.6. Both the method of proxy sources and exploiting translation invariance slightly overestimate the ranks for boxes on the edges of the domain. For example, consider the leaf box in the southwest corner of the domain. This box only sees interactions from the north and east, but the proxy ring also encodes interactions in the south and west. However, the associated costs are negligible compared with the cost of computing factorizations for additional boxes.

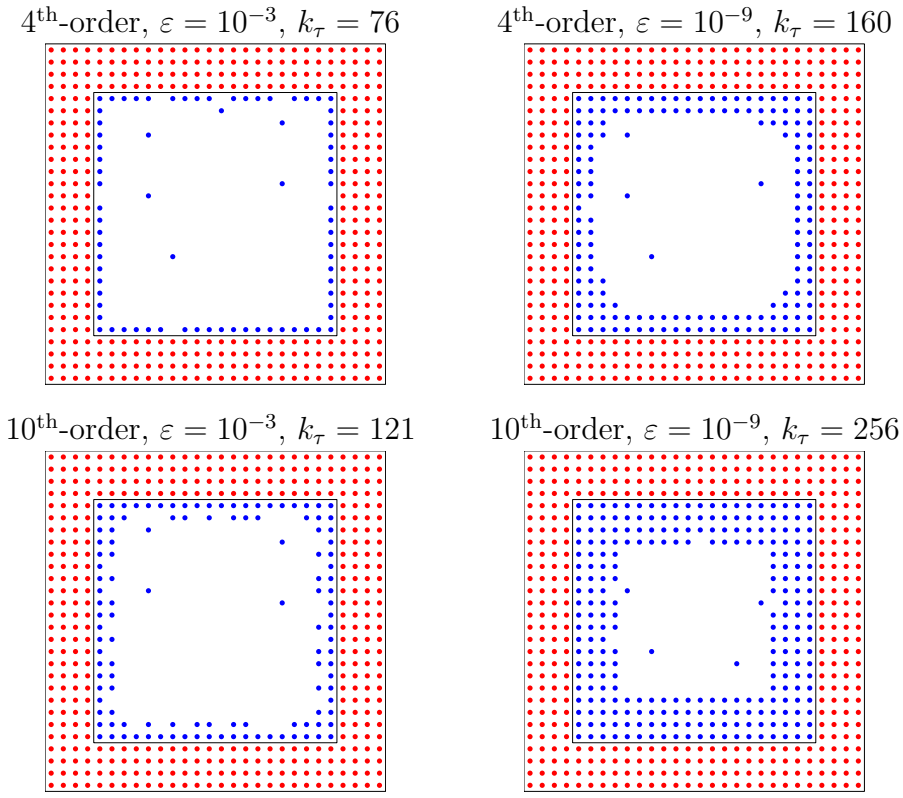


Figure 3.7: Depiction of ID for discretizations with 4^{th}-order (top) and 10^{th}-order (bottom) quadrature corrections at compression tolerances 10^{-3} and 10^{-9}. The skeleton points selected by CPQR are denoted by the blue dots.

Remark 3.7. As mentioned in Remark 3.3, the complexity of our solver depends on the ranks of the various off-diagonal blocks. For a node τ a standard multipole estimate (see, for example, [48]) for area kernels such as (3.6) states that the rank $k_\tau = |\mathcal{I}_\tau^{\text{skel}}|$ (note that this corresponds to the definition of k_τ in Definition 3.2) grows like

$$k_\tau \sim \kappa \times \operatorname{diam} \Omega_\tau. \tag{3.45}$$

When the number of points per wavelength is held constant, (3.45) is equivalent to

$$k_\tau \sim \sqrt{n_\tau} \tag{3.46}$$

where $n_\tau = |\mathcal{I}_\tau|$. In Fig. 3.7 it is shown that increasing the compression tolerance from 10^{-3} to 10^{-9} approximately doubles the rank, so the constants in (3.45) and (3.46) are important to consider for practical efficiency.

Remark 3.8. A minor nuisance in the compression procedure is that while an approximation to the entire operator up to a certain tolerance is sought (cf. (3.39)),

we only control the approximation error locally by specifying the truncation error in making low-rank approximations to individual off-diagonal blocks (cf. (3.41) and (3.43)). Since we are using nested basis matrices, these errors aggregate as we go up the tree and therefore it makes sense to use a tighter compression tolerance on the lower levels of the tree. It is difficult to establish a precise result on what this scaling should be for approximations in the spectral norm, and even if such a result were available, it would likely be impractically pessimistic due to the correlations in the error incurred. In practice we choose the local compression tolerance for a given box to be the global compression tolerance times an estimate of the block's Frobenius norm. While this leads to a slight loss of accuracy as the number of levels in the tree grows, we find this effect negligible in the regimes of interest.

3.4 Direct solver

3.4.1 Description of direct solver

We begin by introducing some notation for the solver. For each node in the tree τ, we define the following vectors in $\mathbb{C}^{n_\tau}$:

$$\mathbf{q}_\tau = \mathbf{q}(\mathcal{I}_\tau), \ \mathbf{w}_\tau = \mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c)\mathbf{q}(\mathcal{I}_\tau^c), \text{ and } \mathbf{f}_\tau = \mathbf{f}(\mathcal{I}_\tau). \tag{3.47}$$

In terms of the physical interpretation introduced in Section 3.3, $\mathbf{q}_\tau$ and $\mathbf{f}_\tau$ represent the restriction of the charges and total field to the points $\{\boldsymbol{x}_j\}_{j\in\mathcal{I}_\tau}$, and $\mathbf{w}_\tau$ represents the incoming field at these points (i.e., the contribution to the total field felt at these points from sources on the points $\{\boldsymbol{x}_j\}_{j\in\mathcal{I}_\tau^c}$). The key observation for accelerated inversion is that for a given non-root node, the contribution to the incoming field from all nodes except for its sibling can be computed on higher levels of the tree, in a similar manner to that in the FMM.

The vectors in (3.47) are of size $\mathcal{O}(N)$ on the higher levels of the tree, and so instead we must work with compressed versions of these quantities. To this end, for each non-root node τ, we define its *outgoing expansion* $\widetilde{\mathbf{q}}_\tau$ and *incoming expansion* $\widetilde{\mathbf{w}}_\tau$ by

$$\widetilde{\mathbf{q}}_\tau = \widehat{\mathbf{V}}_\tau^* \mathbf{q}_\tau \text{ and } \widetilde{\mathbf{w}}_\tau = \widehat{\mathbf{U}}_\tau^\dagger \mathbf{w}_\tau. \tag{3.48}$$

These vectors are only of length k_τ, so there are considerable savings when working with them over the vectors in (3.47) directly.

The outgoing and incoming expansions can be thought of as the generalization of the multipole and local expansions in the FMM. The matrices $\widehat{\mathbf{U}}_\tau$ and $\widehat{\mathbf{V}}_\tau^*$ can then

be interpreted as translation operators. The following lemma provides a formula for the incoming expansions for the children of a non-root node in terms of its incoming expansion and a sibling-sibling interaction.

Lemma 3.3. *Let τ be a non-root parent node with children nodes α and β. If $\widehat{\mathbf{U}}_\alpha$ and $\widehat{\mathbf{U}}_\beta$ have full rank then*

$$\begin{bmatrix} \widetilde{\mathbf{w}}_\alpha \\ \widetilde{\mathbf{w}}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} + \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau. \tag{3.49}$$

Proof. From the definition of the incoming field (3.47), it follows that

$$\begin{bmatrix} \mathbf{w}_\alpha \\ \mathbf{w}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{G}(\mathcal{I}_\alpha, \mathcal{I}_\beta) & \mathbf{G}(\mathcal{I}_\alpha, \mathcal{I}_\tau^c) \\ \mathbf{G}(\mathcal{I}_\beta, \mathcal{I}_\alpha) & \mathbf{0} & \mathbf{G}(\mathcal{I}_\beta, \mathcal{I}_\tau^c) \end{bmatrix} \begin{bmatrix} \mathbf{q}_\alpha \\ \mathbf{q}_\beta \\ \mathbf{q}(\mathcal{I}_\tau^c) \end{bmatrix} \tag{3.50}$$

$$= \begin{bmatrix} \mathbf{0} & \mathbf{G}(\mathcal{I}_\alpha, \mathcal{I}_\beta) \\ \mathbf{G}(\mathcal{I}_\beta, \mathcal{I}_\alpha) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{q}_\alpha \\ \mathbf{q}_\beta \end{bmatrix} + \mathbf{G}(\mathcal{I}_\tau, \mathcal{I}_\tau^c) \mathbf{q}(\mathcal{I}_\tau^c) \tag{3.51}$$

$$= \begin{bmatrix} \mathbf{0} & \mathbf{G}(\mathcal{I}_\alpha, \mathcal{I}_\beta) \\ \mathbf{G}(\mathcal{I}_\beta, \mathcal{I}_\alpha) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{q}_\alpha \\ \mathbf{q}_\beta \end{bmatrix} + \widehat{\mathbf{U}}_\tau \widetilde{\mathbf{w}}_\tau. \tag{3.52}$$

From (3.27) in Definition 3.2 and the definition of the outgoing expansion in (3.47), it follows that the first term in (3.52) can be written

$$\begin{bmatrix} \mathbf{0} & \mathbf{G}(\mathcal{I}_\alpha, \mathcal{I}_\beta) \\ \mathbf{G}(\mathcal{I}_\beta, \mathcal{I}_\alpha) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{q}_\alpha \\ \mathbf{q}_\beta \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{U}}_\alpha & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{U}}_\beta \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{V}}_\alpha^* & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{V}}_\beta^* \end{bmatrix} \begin{bmatrix} \mathbf{q}_\alpha \\ \mathbf{q}_\beta \end{bmatrix} \tag{3.53}$$

$$= \begin{bmatrix} \widehat{\mathbf{U}}_\alpha & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{U}}_\beta \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix}. \tag{3.54}$$

Next, using (3.28) in Definition 3.2, we can write the second term in (3.52) as

$$\widehat{\mathbf{U}}_\tau \widetilde{\mathbf{w}}_\tau = \begin{bmatrix} \widehat{\mathbf{U}}_\alpha & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{U}}_\beta \end{bmatrix} \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau. \tag{3.55}$$

Inserting (3.54) and (3.55) into (3.52) gives us the expression

$$\begin{bmatrix} \mathbf{w}_\alpha \\ \mathbf{w}_\beta \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{U}}_\alpha & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{U}}_\beta \end{bmatrix} \left(\begin{bmatrix} \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} + \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau \right). \tag{3.56}$$

Comparing (3.56) and (3.48), and using the fact that $\widehat{\mathbf{U}}_\alpha$ and $\widehat{\mathbf{U}}_\beta$ are full rank, it is clear that the parenthetical expression in (3.56) must equal the incoming expansions. From this, (3.49) follows. □

We can now describe the solver, which follows the pattern of the solver in [112, Ch. 18], but with accelerations enabled by the structure of the specific coefficient matrix (3.23). First, an upward pass will be made on the tree where certain mathematical operators and a compressed expansion for the right-hand side will be computed. Then at the root level, the computed operators will be used to compute the incoming expansions for both the root node's children. A downward pass is then made to compute the incoming expansions for all of the remaining nodes on the tree. Finally, at the level of the leaves, the solution vector is assembled.

We begin by considering an arbitrary leaf node on the tree τ. Let $\mathbf{B}_\tau = \mathbf{B}(\mathcal{I}_\tau, \mathcal{I}_\tau)$. Restricting the system (3.23) to the rows indexed by $\mathcal{I}_\tau$ and using the notation in (3.47), gives us the equation

$$(\mathbf{I} + \mathbf{B}_\tau \mathbf{G}_\tau)\mathbf{q}_\tau + \mathbf{B}_\tau \mathbf{w}_\tau = \mathbf{f}_\tau. \tag{3.57}$$

We then let

$$\mathbf{X}_\tau = (\mathbf{I} + \mathbf{B}_\tau \mathbf{G}_\tau)^{-1}, \tag{3.58}$$

multiply (3.57) through by $\mathbf{X}_\tau$, and use the fact that $\mathbf{w}_\tau = \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau$ to get

$$\mathbf{q}_\tau + \mathbf{X}_\tau \mathbf{B}_\tau \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau = \mathbf{r}_\tau \tag{3.59}$$

where

$$\mathbf{r}_\tau = \mathbf{X}_\tau \mathbf{f}_\tau. \tag{3.60}$$

From (3.59) it is clear that after the incoming expansion $\widetilde{\mathbf{w}}_\tau$ is determined, the solution vector $\mathbf{q}_\tau$ can be immediately determined. Now, multiplying (3.59) through by $\mathbf{V}_\tau^*$ and letting

$$\mathbf{S}_\tau = \mathbf{V}_\tau^* \mathbf{X}_\tau \mathbf{B}_\tau \mathbf{U}_\tau \tag{3.61}$$

and

$$\widetilde{\mathbf{r}}_\tau = \mathbf{V}_\tau^* \mathbf{r}_\tau \tag{3.62}$$

produces the equilibrium equation

$$\widetilde{\mathbf{q}}_\tau + \mathbf{S}_\tau \widetilde{\mathbf{w}}_\tau = \widetilde{\mathbf{r}}_\tau. \tag{3.63}$$

Note that whereas (3.57) contains n_τ equations, (3.63) only contains k_τ equations.

The matrix $\mathbf{X}_\tau$ in (3.58) is a *local solution operator*. That is, it is the operator that maps a total field restricted to the points $\{\boldsymbol{x}_j\}_{j \in \mathcal{I}_\tau}$ to the strengths of the charges at those points that generate that field in the absence of an incoming field. Similarly, the matrix $\mathbf{S}_\tau$ is a *scattering matrix* which maps an incoming expansion to a corresponding

total field in the absence of charges on $\{\boldsymbol{x}_j\}_{j\in\mathcal{I}_\tau}$. This procedure is repeated, computing for each leaf τ the vectors $\mathbf{r}_\tau$ and $\widetilde{\mathbf{r}}_\tau$, and the matrices $\mathbf{X}_\tau$ and $\mathbf{S}_\tau$ such that (3.63) holds.

We now move one level up the tree, and consider a node τ with children that are leaves α and β. Since (3.63) holds for α and β, we have from the previous step the equations

$$\begin{aligned}\widetilde{\mathbf{q}}_\alpha + \mathbf{S}_\alpha \widetilde{\mathbf{w}}_\alpha &= \widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{q}}_\beta + \mathbf{S}_\beta \widetilde{\mathbf{w}}_\beta &= \widetilde{\mathbf{r}}_\beta.\end{aligned} \tag{3.64}$$

Using Lemma 3.3, we can insert (3.49) for the incoming expansions in (3.64), which produces

$$\begin{bmatrix}\widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta\end{bmatrix} + \begin{bmatrix}\mathbf{S}_\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_\beta\end{bmatrix}\left(\begin{bmatrix}\mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta\end{bmatrix} + \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau\right) = \begin{bmatrix}\widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta\end{bmatrix}. \tag{3.65}$$

Letting

$$\mathbf{X}_\tau = \begin{bmatrix}\mathbf{I} & \mathbf{S}_\alpha \mathbf{G}_{\alpha,\beta} \\ \mathbf{S}_\beta \mathbf{G}_{\beta,\alpha} & \mathbf{I}\end{bmatrix}^{-1} \tag{3.66}$$

and multiplying (3.67) by $\mathbf{X}_\tau$ leads to

$$\begin{bmatrix}\widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta\end{bmatrix} + \mathbf{X}_\tau \begin{bmatrix}\mathbf{S}_\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_\beta\end{bmatrix}\mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau = \mathbf{X}_\tau \begin{bmatrix}\widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta\end{bmatrix}. \tag{3.67}$$

Defining

$$\mathbf{S}_\tau = \mathbf{V}_\tau^* \mathbf{X}_\tau \begin{bmatrix}\mathbf{S}_\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_\beta\end{bmatrix}\mathbf{U}_\tau \tag{3.68}$$

and

$$\widetilde{\mathbf{r}}_\tau = \mathbf{V}_\tau^* \mathbf{X}_\tau \begin{bmatrix}\widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta\end{bmatrix}, \tag{3.69}$$

and then multiplying (3.67) through by $\mathbf{V}_\tau^*$ then yields

$$\widetilde{\mathbf{q}}_\tau + \mathbf{S}_\tau \widetilde{\mathbf{w}}_\tau = \widetilde{\mathbf{r}}_\tau. \tag{3.70}$$

Again, $\mathbf{X}_\tau$ in (3.66) and $\mathbf{S}_\tau$ in (3.68) can be interpreted as a local solution operator and scattering matrix, respectively. Here we have started with $k_\alpha + k_\beta$ equations in (3.64) and compressed down to k_τ equations in (3.70). We then repeat the process for all other nodes on level $L-1$. Since (3.70) has the same form as (3.63), we can repeat this process at level $L-2$ and then continue up the tree one level at a time, all the way up to level 1. At each level, we start with equations of the form (3.64), and then compute the quantities in (3.66), (3.68), and (3.70).

Now, let τ denote the root node and let α and β denote its children. Since the root node of the tree corresponds to the whole domain Ω, there is no associated incoming field. Thus, the analog of (3.49) is simply

$$\begin{bmatrix} \widetilde{\mathbf{w}}_\alpha \\ \widetilde{\mathbf{w}}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix}, \tag{3.71}$$

which means the analog of (3.67) is

$$\begin{bmatrix} \mathbf{I} & \mathbf{S}_\alpha \mathbf{G}_{\alpha,\beta} \\ \mathbf{S}_\beta \mathbf{G}_{\beta,\alpha} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta \end{bmatrix}. \tag{3.72}$$

The linear system in (3.72) is directly inverted to obtain the outgoing expansions $\widetilde{\mathbf{q}}_\alpha$ and $\widetilde{\mathbf{q}}_\beta$, from which the incoming expansions on α and β are obtained using (3.71).

We now move one level down the tree. Let the node τ be a child of the root node, and let α and β denote the children of τ. From the previous step, we have computed the incoming expansion $\widetilde{\mathbf{w}}_\tau$. Using (3.67), the outgoing expansions $\widetilde{\mathbf{q}}_\alpha$ and $\widetilde{\mathbf{q}}_\beta$ are computed. The incoming expansions $\widetilde{\mathbf{w}}_\alpha$ and $\widetilde{\mathbf{w}}_\beta$ are then computed using (3.49). The other child of the root is analogously processed. We then proceed down the tree, one level at a time, computing the outgoing expansions of the children of each node using (3.67) and then the incoming expansions using (3.49). We continue this until we reach the leaf level. Finally, for each leaf node τ, the solution $\mathbf{q}_\tau$ is assembled using (3.59).

In the above discussion we have not distinguished between the procedure for computing the inverse of the HBS matrix from that for applying the inverse to a right-hand side, which we call the *build stage* and *apply stage*, respectively. We can separate the two by letting the build stage consist of a single upward pass on the tree where the local solution operators and scattering matrices are computed using (3.58), (3.61), (3.66), and (3.68). Pseudocode for this procedure is given in Algorithm 3.1. After these operators have been computed, the apply stage consists of an upward pass to compute the compressed right-hand sides using (3.60), (3.62), and (3.69), followed by a root level solve where the outgoing and incoming expansions on the children of the root node are determined by solving (3.72) and (3.71), then a downward pass where the outgoing expansions and incoming expansions are computed using (3.67) and (3.49), and finally the solution is computed using (3.59). This is summarized in Algorithm 3.2.

We will see in Section 3.4.2 that the dominant cost in the solver is the dense inversion required to compute the local solution operators in (3.66) for the root level of the tree. This can be accelerated by observing that the matrices $\mathbf{G}_{\alpha,\beta}$ and $\mathbf{G}_{\beta,\alpha}$ that appear in the off-diagonal blocks of (3.66) encode the sibling-sibling interactions

between two adjacent boxes. Since these boxes share only a single side, we expect the numerical ranks of these matrices to be only about 25% of their sizes. Thus, we can compute the low-rank factorizations $\mathbf{G}_{\alpha,\beta} = \mathbf{L}_{\alpha,\beta}\mathbf{R}^*_{\alpha,\beta}$ and $\mathbf{G}_{\beta,\alpha} = \mathbf{L}_{\beta,\alpha}\mathbf{R}^*_{\beta,\alpha}$, which only needs to be computed once per level by translation invariance, and then use the Woodbury formula to compute $\mathbf{X}_\tau$ using the expression

$$\mathbf{X}_\tau = \mathbf{I} - \begin{bmatrix} \mathbf{S}_\alpha \mathbf{L}_{\alpha,\beta} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_\beta \mathbf{L}_{\beta,\alpha} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{R}^*_{\beta,\alpha}\mathbf{L}_{\beta,\alpha} \\ \mathbf{R}^*_{\alpha,\beta}\mathbf{L}_{\alpha,\beta} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} & \mathbf{R}^*_{\alpha,\beta} \\ \mathbf{R}^*_{\beta,\alpha} & \mathbf{0} \end{bmatrix}. \tag{3.73}$$

This substantially reduces the dimensions of the matrix that must be inverted. In spite of not changing the asymptotic complexity, this substantially reduces the constants.

Algorithm 3.1 Build inverse as described in Section 3.4.1

Input: HBS compression of coefficient matrix specified by $(\mathbf{U}_\tau, \mathbf{V}_\tau)$ for all non-root nodes, $(\mathbf{B}_\tau, \mathbf{G}_\tau)$ for all leaves, and $(\mathbf{G}_{\alpha,\beta}, \mathbf{G}_{\beta,\alpha})$ for all parents with children α and β
Output: Inverse of HBS matrix specified by $\mathbf{X}_\tau$ for all nodes and $\mathbf{S}_\tau$ for all non-root nodes

for all τ a leaf **do**
  $\mathbf{X}_\tau = (\mathbf{I} + \mathbf{B}_\tau \mathbf{G}_\tau)^{-1}$
  $\mathbf{S}_\tau = \mathbf{V}_\tau^* \mathbf{X}_\tau \mathbf{B}_\tau \mathbf{U}_\tau$
end for
for $\ell = L-1, L-3, \ldots, 0$ **do**
  for all τ on level ℓ **do**
    $[\alpha, \beta] \leftarrow$ children of τ
    $\mathbf{X}_\tau = \begin{bmatrix} \mathbf{I} & \mathbf{S}_\alpha \mathbf{G}_{\alpha,\beta} \\ \mathbf{S}_\beta \mathbf{G}_{\beta,\alpha} & \mathbf{I} \end{bmatrix}^{-1}$
    if $\ell \neq 0$ **then**
      $\mathbf{S}_\tau = \mathbf{V}_\tau^* \mathbf{X}_\tau \begin{bmatrix} \mathbf{S}_\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_\beta \end{bmatrix} \mathbf{U}_\tau$
    end if
  end for
end for

3.4.2 Complexity estimate

A complexity estimate can be given for each stage of our solver using the rank growth assumption in Remark 3.7, which states that for a node τ the rank $k_\tau = |\mathcal{I}_\tau^{\text{skel}}|$ scales like

$$k_\tau = \mathcal{O}_{\kappa,\varepsilon}(\sqrt{n_\tau}) \tag{3.74}$$

where $n_\tau = |\mathcal{I}_\tau|$ is the number of points in the box Ω_τ. Since we have used a uniform division in space and translation invariance in Section 3.2.4, k_τ and n_τ depend only

Algorithm 3.2 Apply inverse to right-hand side $\mathbf{f}$ as described in Section 3.4.1

Input: HBS matrix, factors for inverse outputted by Algorithm 3.1, and right-hand side $\mathbf{f}$

Output: Solution vector $\mathbf{q}$

// Upwards pass
for all τ a leaf **do**
    $\mathbf{r}_\tau = \mathbf{X}_\tau \mathbf{f}_\tau$
    $\widetilde{\mathbf{r}}_\tau = \mathbf{V}_\tau^* \mathbf{r}_\tau$
end for
for $\ell = L-1, L-3, \ldots, 1$ **do**
    for all τ on level ℓ **do**
        $[\alpha, \beta] \leftarrow$ children of τ
        $\widetilde{\mathbf{r}}_\tau = \mathbf{V}_\tau^* \mathbf{X}_\tau \begin{bmatrix} \widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta \end{bmatrix}$
    end for
end for

// Top-level solve
$\tau \leftarrow$ root
$[\alpha, \beta] \leftarrow$ children of τ
$$\begin{bmatrix} \widetilde{\mathbf{w}}_\alpha \\ \widetilde{\mathbf{w}}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \mathbf{X}_\tau \begin{bmatrix} \widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta \end{bmatrix}$$

// Downwards pass
for $\ell = 2, 3, \ldots, L-1$ **do**
    for all τ on level ℓ **do**
        $[\alpha, \beta] \leftarrow$ children of τ
        $$\begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} = \mathbf{X}_\tau \left(\begin{bmatrix} \widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta \end{bmatrix} - \begin{bmatrix} \mathbf{S}_\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_\beta \end{bmatrix} \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau \right)$$
        $$\begin{bmatrix} \widetilde{\mathbf{w}}_\alpha \\ \widetilde{\mathbf{w}}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} + \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau$$
    end for
end for

// Form solution vector
for all τ a leaf **do**
    $\mathbf{q}_\tau = \mathbf{r}_\tau - \mathbf{X}_\tau \mathbf{B}_\tau \mathbf{U}_\tau \widetilde{\mathbf{w}}_\tau$
end for

on the level and thus the quantities $k_\ell = k_\tau$ and $n_\ell = n_\tau$ for an arbitrary node τ on level ℓ are uniquely defined. Similarly, the quantity $|\mathcal{I}_\ell^{\text{proxy}}|$ is well-defined. We note that the relationship

$$n_\ell = \frac{N}{2^\ell}, \quad \ell = 0, 1, \ldots, L \tag{3.75}$$

holds. For the compression stage of the algorithm, we need to compute (3.41) once on the leaf level and compute (3.43) once per level on the levels $1, 2, \ldots, L-1$. The method of proxy sources allows us to compute a low-rank factorization of a matrix that is $n_\ell \times |\mathcal{I}_\ell^{\text{proxy}}|$ for a node on level $\ell = L$ and $2k_{\ell+1} \times |\mathcal{I}_\ell^{\text{proxy}}|$ for nodes on levels $\ell = 1, 2, \ldots, L-1$. Since the proxy ring scales with the perimeter of the box we know that

$$|\mathcal{I}_\ell^{\text{proxy}}| = \mathcal{O}_{\kappa,\varepsilon}(\sqrt{n_\ell}). \tag{3.76}$$

Combining these observations with the fact that a rank-k ID on an $m \times n$ matrix using CPQR takes $\mathcal{O}(mnk)$ operations gives us the following complexity estimate for the compression stage:

$$t_{\text{comp}} \sim n_L \sqrt{n}_L k_L + \sum_{\ell=1}^{L-1} 2k_{\ell+1}\sqrt{n_\ell} k_\ell \tag{3.77}$$

$$\sim n_L^2 + \sum_{\ell=1}^{L-1} 2n_\ell \sqrt{n_{\ell+1}} \tag{3.78}$$

$$\sim \frac{N^2}{2^{2L}} + \sum_{\ell=1}^{L-1} \frac{N^{3/2}}{2^{(3\ell-1)/2}} \tag{3.79}$$

$$\sim N_{\text{dense}}^2 + \sum_{\ell=1}^{L-1} \frac{N^{3/2}}{2^{(3\ell-1)/2}} \tag{3.80}$$

$$\sim N^{3/2}, \tag{3.81}$$

where we have used in (3.80) the fact that there are about N_{dense} points in a box on the leaf level by the construction in Section 3.3. It is also clear from (3.80) that the computation is dominated by the higher levels of the tree.

As discussed in Section 3.4.1, the inversion stage consists of computing the local solution operators $\mathbf{X}_\tau$ and scattering matrices $\mathbf{S}_\tau$. From the formulas (3.58), (3.61), (3.66), and (3.68), it is clear that computing $\mathbf{X}_\tau$ is more expensive than $\mathbf{S}_\tau$ for any node τ. On the leaf level computing $\mathbf{X}_\tau$ requires dense inversion of an $n_L \times n_L$ matrix if τ is a leaf and a $2k_{\ell+1} \times 2k_{\ell+1}$ matrix if τ is on level $\ell = 0, 1, \ldots, L-1$. Using the

fact that there are 2^ℓ boxes on level ℓ yields the estimate

$$t_{\text{build}} \sim 2^L n_L^3 + \sum_{\ell=0}^{L-1} (2k_{l+1})^3 2^\ell \tag{3.82}$$

$$\sim 2^L n_L^3 + \sum_{\ell=0}^{L-1} 2^{\ell+3} n_{\ell+1}^{3/2} \tag{3.83}$$

$$\sim 2^L \left(\frac{N}{2^L}\right)^3 + \sum_{\ell=0}^{L-1} 2^{\ell+3} \left(\frac{N}{2^{\ell+1}}\right)^{3/2} \tag{3.84}$$

$$\sim 2^L N_{\text{dense}}^3 + \sum_{\ell=0}^{L-1} \frac{N^{3/2}}{2^{(\ell-5)/2}} \tag{3.85}$$

$$\sim N^{3/2}, \tag{3.86}$$

where again the cost of processing the highest levels of the tree dominates.

The dominant cost of applying the inverse is matrix-vector multiplication with matrices that are $n_L \times n_L$ on the leaf level and $2k_{\ell+1} \times 2k_{\ell+1}$ on level $\ell = 0, 1, \ldots, L-1$. Thus, the cost of the apply stage can be estimated by

$$t_{\text{apply}} \sim 2^L n_L^2 + \sum_{\ell=0}^{L-1} 2^\ell (2k_{\ell+1})^2 \tag{3.87}$$

$$\sim 2^L n_L^2 + \sum_{\ell=0}^{L-1} 2^{\ell+2} n_{\ell+1} \tag{3.88}$$

$$\sim 2^L n_L^2 + \sum_{\ell=0}^{L-1} 2^{\ell+2} \frac{N}{2^{\ell+1}} \tag{3.89}$$

$$\sim 2^L N_{\text{dense}}^2 + \sum_{\ell=0}^{L-1} 2N \tag{3.90}$$

$$\sim N \log N, \tag{3.91}$$

where in the last step we have used the fact that the tree has $\log N$ levels. This analysis shows that unlike the other two stages, the cost of the apply stage is approximately balanced equally among all levels of the tree.

Remark 3.9. In the regime where κ is fixed and N must be increased, it is in principle possible to accelerate the compression and build stages of the solver to $\mathcal{O}(N)$ complexity using nested trees [45] or hierarchical interpolative factorization [89]. However, both of these methods have increased constants, and it is currently under investigation if this trade-off results in an increase in practical efficiency in our regimes of interest.

Remark 3.10. In the case where the wavenumber κ is increased and the number of points per wavelength is fixed so that $N \sim \kappa^2$, our solver has $\mathcal{O}(\kappa^3)$ complexities in the compression and build stages, and $\mathcal{O}(\kappa^2 \log \kappa)$ complexity in the apply stage. While there exist subcubic solvers in special cases, such as when the scattering potential is radially symmetric (see e.g., [27, 93]), we are not aware of any methods faster than $\mathcal{O}(\kappa^3)$ for the general case.

3.4.3 Comparison to other direct solvers

Two closely related solvers to the one presented here are the Woodbury-based solver in [62] and the sparse matrix embedding solver in [88]. The relation between our solver and those can be elucidated by considering an HBS matrix with two levels and considering the sparse matrix embedding introduced in [88]. In our notation this yields the linear system

$$\begin{bmatrix} (\mathbf{I} + \mathbf{B}_\alpha \mathbf{G}_\alpha) & \mathbf{0} & \mathbf{B}_\alpha \mathbf{U}_\alpha & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\mathbf{I} + \mathbf{B}_\beta \mathbf{G}_\beta) & \mathbf{0} & \mathbf{B}_\beta \mathbf{U}_\beta & \mathbf{0} & \mathbf{0} \\ -\mathbf{V}_\alpha^* & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{V}_\beta^* & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{q}_\alpha \\ \mathbf{q}_\beta \\ \widetilde{\mathbf{w}}_\alpha \\ \widetilde{\mathbf{w}}_\beta \\ \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{f}_\alpha \\ \mathbf{f}_\beta \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{3.92}$$

The approach advocated in [88] is to use a sparse direct solver, such as UMFPACK, directly on the system (3.92). The motivation here is that the pivots can be adaptively chosen to reduce fill-in for the purpose of efficiency and to avoid the inversion of ill-conditioned matrices for the purpose of stability. In practice we find that this approach is stable, but, without careful tuning of the sparse solver, can have large constants in the asymptotic complexities for memory and running time.

On the other hand, the upward pass of the solvers in [62] and Section 3.4.1 can be viewed as a fixed choices of pivots. This can be seen by first eliminating the contributions from $\mathbf{q}_\alpha$ and $\mathbf{q}_\beta$ by using the $(1,1)$ and $(2,2)$ blocks to eliminate the $(3,1)$ and $(4,2)$ blocks in (3.92). This results in the system

$$\begin{bmatrix} \mathbf{S}_\alpha & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_\beta & \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{0} & -\mathbf{I} & \mathbf{G}_{\beta,\alpha} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{w}}_\alpha \\ \widetilde{\mathbf{w}}_\beta \\ \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{3.93}$$

The system solved at the root level in the solver in [62] can then be derived by eliminating the $(3,1)$ and $(4,2)$ blocks of (3.93), which yields the system

$$\begin{bmatrix} \mathbf{S}_\alpha^{-1} & \mathbf{G}_{\alpha,\beta} \\ \mathbf{G}_{\beta,\alpha} & \mathbf{S}_\beta^{-1} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{S}_\alpha^{-1} \widetilde{\mathbf{r}}_\alpha \\ \mathbf{S}_\beta^{-1} \widetilde{\mathbf{r}}_\beta \end{bmatrix}. \tag{3.94}$$

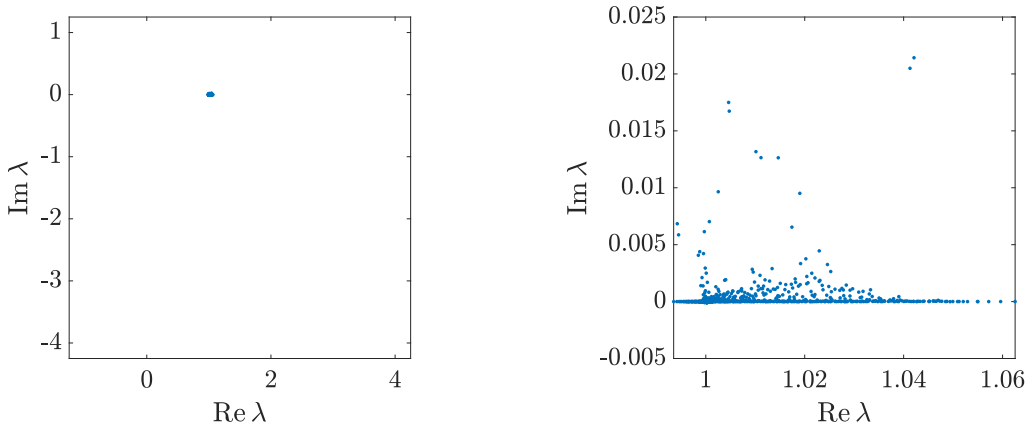


Figure 3.8: Spectrum of preconditioned operator for lens problem described in Section 3.2.3. The right plot is a zoomed in version of the left plot.

Inverting the matrix in (3.94) requires inverting the scattering matrices $\mathbf{S}_\alpha$ and $\mathbf{S}_\beta$. These matrices need not be well-conditioned, and looking at (3.61), we actually anticipate ill-conditioning when b obtains values that are small in magnitude in Ω_α or Ω_β. In fact, for the examples in Section 3.6, we frequently encounter scattering matrices with condition numbers greater than the reciprocal of machine accuracy in double precision arithmetic, which would then result in a substantial loss of accuracy.

In contrast, the approach presented in Section 3.4.1 corresponds to eliminating the $(1,1)$ and $(2,2)$ blocks in (3.93), which results in the system

$$\begin{bmatrix} \mathbf{I} & \mathbf{S}_\alpha \mathbf{G}_{\alpha,\beta} \\ \mathbf{S}_\beta \mathbf{G}_{\beta,\alpha} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{q}}_\alpha \\ \widetilde{\mathbf{q}}_\beta \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{r}}_\alpha \\ \widetilde{\mathbf{r}}_\beta \end{bmatrix}. \tag{3.95}$$

Indeed, (3.95) appears as (3.72). In this formulation $\mathbf{S}_\alpha$ and $\mathbf{S}_\beta$ do not need to be inverted, avoiding the instability in the Woodbury-based scheme and reducing the amount of dense linear algebra necessary. Furthermore, we empirically find that this approach is equally stable to the one in [88] and considerably more efficient.

3.5 Using the solver as a preconditioner

If high accuracy is needed in the computed solution, the compression tolerance must be tightened and high-order Duan–Rokhlin quadratures must be used. Both of these increase the ranks of the off-diagonal block in the HBS format and therefore slow down each stage of the solution procedure. Thus, it is often preferable to use the direct solver to compute a coarse approximation to the inverse of the coefficient matrix

and then use this as a preconditioner for an iterative solver, such as the one described in Section 3.2.3.

For example, suppose we wanted to solve a system of the form (3.23) given by

$$(\mathbf{I} + \mathbf{BG} + \mathbf{BT}^{(10)})\mathbf{q} = \mathbf{f} \tag{3.96}$$

where $\mathbf{T}^{(10)}$ denotes the matrix containing 10^{th}-order Duan–Rokhlin corrections. For a preconditioner we use the direct solver to obtain a data-sparse approximation $\mathbf{M}$ such that

$$\mathbf{M} \approx (\mathbf{I} + \mathbf{BG} + \mathbf{BT}^{(4)})^{-1} \tag{3.97}$$

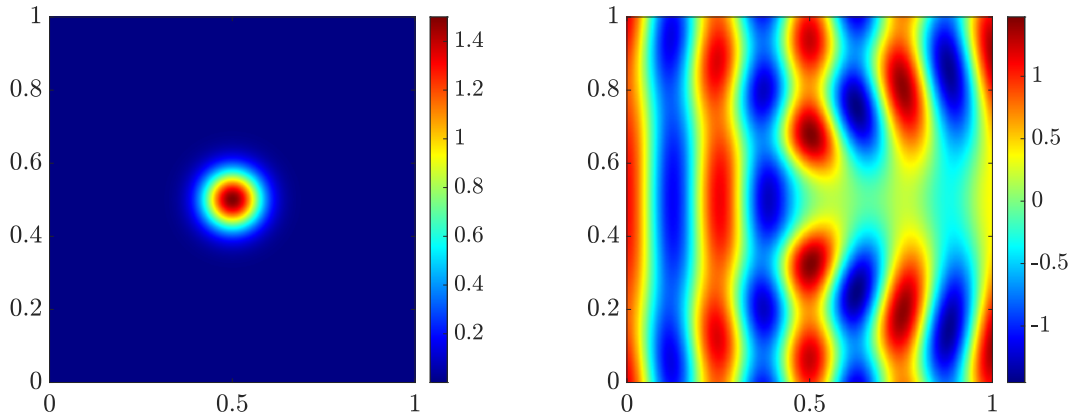
where $\mathbf{T}^{(4)}$ is now a diagonal matrix with the 4^{th}-order Duan–Rokhlin corrections. After such an $\mathbf{M}$ is computed, the system

$$\mathbf{M}(\mathbf{I} + \mathbf{BG} + \mathbf{BT}^{(10)})\mathbf{q} = \mathbf{Mf} \tag{3.98}$$

can then be solved using GMRES, where the sparsity of $\mathbf{B}$, the Toeplitz structure of $\mathbf{G} + \mathbf{T}^{(10)}$, and Algorithm 3.2 are exploited to apply the coefficient matrix to a vector in quasilinear complexity. It is empirically observed that computing $\mathbf{M}$ to only a few digits is sufficient to reduce the number of iterations to a feasible number in a broad range of cases. This is illustrated in Fig. 3.8, which plots the spectrum of the preconditioned operator, when the aforementioned approach is applied with compression tolerance 10^{-2}, corresponding to the lens problem described at the end of Section 3.2.3 (cf. Fig. 3.3(d)). The number of iterations required to attain a residual with norm below 10^{-10} decreases from 79 to 6. Computing the preconditioner only takes a fraction of a second on a laptop, and the time per iteration when the preconditioner is used only increases by a factor of 2, resulting in an overall speedup for even this small problem.

3.6 Numerical results

In this section we illustrate the efficacy of our direct solver as a means of both inversion and preconditioning. All experiments were done using a workstation with 2 Intel Xeon Gold 6254 processors (18 cores at 3.1GHz base frequency) with 768 GB of RAM, with all codes implemented in MATLAB 2019b.



(a) (b)

Figure 3.9: (a) Gaussian bump scattering potential. (b) The real part of the total field when a plane wave $\kappa = 25$ impinges on the Gaussian bump

3.6.1 Direct solution for a Gaussian bump

Our first set of experiments study the scaling of the direct solver as the mesh is uniformly refined with 10^{th}-order Duan–Rokhlin quadratures[2] at different compression tolerances in the low-frequency regime with $\kappa = 25$, so that the computational domain is about 4 wavelengths wide, where the plane wave

$$u_{\text{inc}}(x_1, x_2) = \exp(i\kappa(x_1 - 0.5)) \tag{3.99}$$

impinges on a Gaussian bump scattering potential given by

$$b(x_1, x_2) = 1.5 \exp(-160\sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}) \tag{3.100}$$

and plotted Fig. 3.9(a). We tabulate the following quantities when the compression tolerance is 10^{-3}, 10^{-6}, 10^{-9}, and 10^{-12} in Tables 3.2, 3.3, 3.4, and 3.5, respectively:

N	The total number of degrees of freedom
h	The mesh width. Note that $h = 1/\sqrt{N}$
t_{comp}	Time in seconds to compute the HBS approximation
t_{build}	Time in seconds to invert the HBS approximation
t_{apply}	Time in seconds to apply the inverse a single time
mem	Total memory in gigabytes used to store both the HBS approximation and its inverse
res	The relative residual error in the computed solution

[2]The codes to generate the Duan–Rokhlin quadratures were generously provided by Ran Duan and Vladimir Rokhlin.

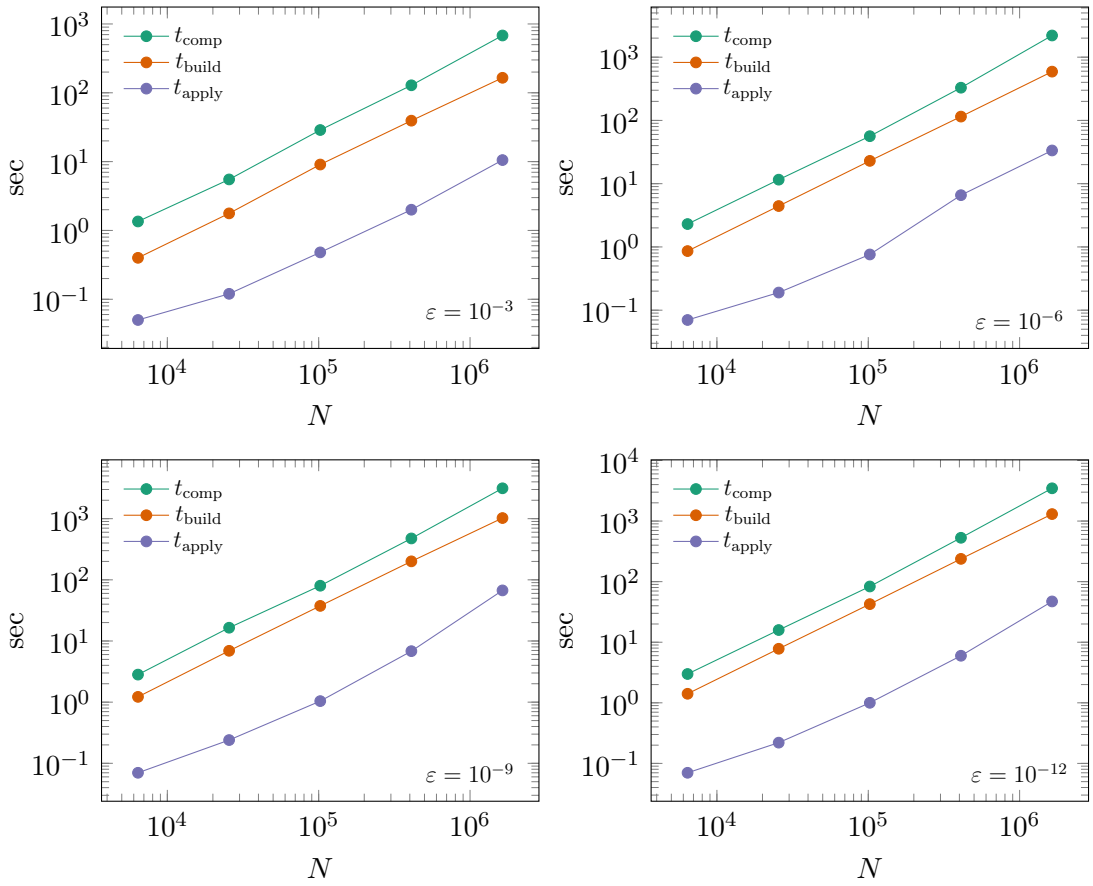


Figure 3.10: Scaling data for Section 3.6.1

N	h	t_{comp}	t_{build}	t_{apply}	mem	res
6400	0.0125	1.35	0.40	0.05	0.12	$7.74 \cdot 10^{-5}$
25600	0.00625	5.51	1.77	0.12	0.60	$5.71 \cdot 10^{-5}$
102400	0.003125	28.78	9.07	0.48	2.94	$9.75 \cdot 10^{-5}$
409600	0.0015625	128.50	39.35	2.00	12.14	$1.56 \cdot 10^{-4}$
1638400	0.00078125	680.04	165.15	10.53	48.59	$5.77 \cdot 10^{-4}$

Table 3.2: Data for Section 3.6.1 with compression tolerance 10^{-3}

N	h	t_{comp}	t_{build}	t_{apply}	mem	res
6400	0.0125	2.30	0.86	0.07	0.27	$9.54 \cdot 10^{-9}$
25600	0.00625	11.57	4.43	0.19	1.42	$7.13 \cdot 10^{-8}$
102400	0.003125	56.32	22.93	0.76	6.86	$4.15 \cdot 10^{-8}$
409600	0.0015625	328.92	114.94	6.58	31.33	$4.39 \cdot 10^{-8}$
1638400	0.00078125	2201.34	590.61	33.50	145.71	$3.14 \cdot 10^{-7}$

Table 3.3: Data for Section 3.6.1 with compression tolerance 10^{-6}

N	h	t_{comp}	t_{build}	t_{apply}	mem	res
6400	0.0125	2.82	1.22	0.07	0.36	$1.57 \cdot 10^{-12}$
25600	0.00625	16.48	6.91	0.24	1.98	$3.37 \cdot 10^{-12}$
102400	0.003125	80.08	37.49	1.04	10.01	$1.45 \cdot 10^{-11}$
409600	0.0015625	475.07	200.59	6.82	48.22	$3.48 \cdot 10^{-9}$
1638400	0.00078125	3139.07	1023.81	67.17	225.26	$2.99 \cdot 10^{-9}$

Table 3.4: Data for Section 3.6.1 with compression tolerance 10^{-9}

N	h	t_{comp}	t_{build}	t_{apply}	mem	res
6400	0.0125	2.99	1.41	0.07	0.38	$1.87 \cdot 10^{-15}$
25600	0.00625	15.90	7.80	0.22	2.11	$3.80 \cdot 10^{-15}$
102400	0.003125	83.22	42.56	1.00	10.82	$6.94 \cdot 10^{-15}$
409600	0.0015625	529.54	237.64	5.98	52.79	$1.71 \cdot 10^{-14}$
1638400	0.00078125	3474.55	1304.09	47.13	250.27	$5.07 \cdot 10^{-14}$

Table 3.5: Data for Section 3.6.1 with compression tolerance 10^{-12}

The scaling of t_{comp}, t_{build}, and t_{apply} with N is further illustrated in Fig. 3.10, and the total field is plotted in Fig. 3.9(b). The slopes of the lines in Fig. 3.10 closely match the predictions based on the analysis in Section 3.4.2, and the complexities appear to scale somewhere between $\mathcal{O}(\log(1/\varepsilon))$ and $\mathcal{O}(\log(1/\varepsilon)^2)$, where ε is the compression tolerance, in all cases. It is interesting to note that for these experiments, the residual error increases with the number of degrees of freedom; this is due to the effect mentioned in Remark 3.8. However, in all cases, the norm of the residual is close to the compression tolerance.

3.6.2 Direct solution for a cavity

As mentioned in Section 3.2.3, iterative solvers excel in environments where the scattering potentials produce little to no backscattering, such as the one in the previous section. In this section we will consider scattering by a highly refractive cavity with scattering potential given in polar coordinates by

$$b(r, \theta) = (1 - \sin(0.5\theta)^{500}) \exp(-2000(0.1 - r^2)^2), \tag{3.101}$$

which is shown in Fig. 3.11(a). The incident field will be the same plane wave (3.99), and the wavenumber will be taken to be $\kappa = 50.27$, so that the domain is about 8 wavelengths wide. We again study the scaling of the solver when 10^{th}-order quadratures

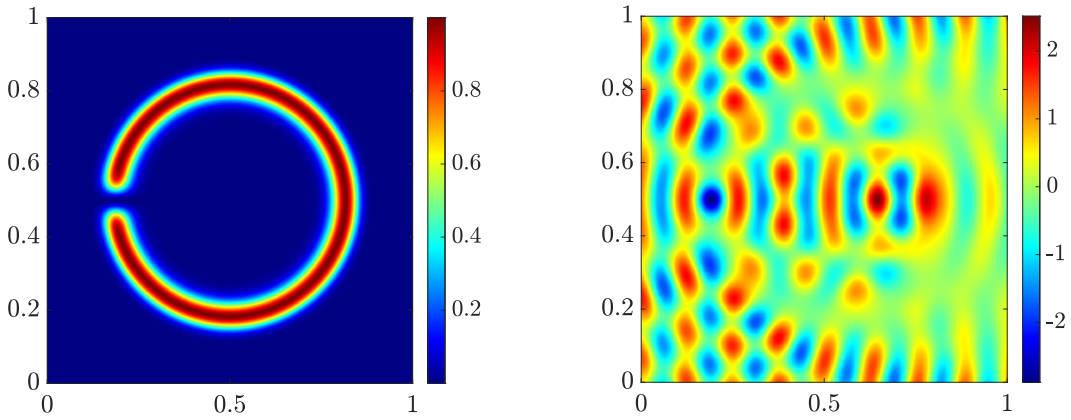


Figure 3.11: (a) Cavity scattering potential. (b) The real part of the total field when a plane wave with $\kappa = 50.27$ impinges on the cavity

are used and the number of degrees of freedom is increased. We record the same quantities as in Section 3.6.1 in Tables 3.6, 3.7, 3.8, and 3.9 when the compression tolerance is 10^{-3}, 10^{-6}, 10^{-9}, and 10^{-12}, respectively. The scaling of the solver is illustrated by Fig. 3.12.

The real part of the total field computed by the solver is plotted in Fig. 3.11(b). Besides a slight increase in computational cost across the board, attributable to the increased wavenumber, the results are in line with those in the case for the Gaussian bump. This illustrates how the direct solver is agnostic to features of the scattering potential.

N	h	t_{comp}	t_{build}	t_{apply}	mem	res
6400	0.0125	2.21	0.46	0.05	0.15	$6.42 \cdot 10^{-4}$
25600	0.00625	5.36	2.16	0.14	0.67	$2.82 \cdot 10^{-4}$
102400	0.003125	30.70	9.63	0.77	3.08	$3.68 \cdot 10^{-4}$
409600	0.0015625	167.84	46.80	3.09	14.52	$6.86 \cdot 10^{-4}$
1638400	0.00078125	940.77	222.96	13.16	65.10	$1.14 \cdot 10^{-3}$

Table 3.6: Data for Section 3.6.2 with compression tolerance 10^{-3}

N	h	t_{comp}	t_{build}	t_{apply}	mem	res
6400	0.0125	2.42	0.94	0.07	0.30	$9.52 \cdot 10^{-8}$
25600	0.00625	12.59	4.97	0.42	1.57	$5.89 \cdot 10^{-8}$
102400	0.003125	60.27	25.71	1.12	7.64	$5.54 \cdot 10^{-7}$
409600	0.0015625	360.95	130.26	6.52	35.64	$5.75 \cdot 10^{-7}$
1638400	0.00078125	2280.63	645.20	25.84	158.14	$1.09 \cdot 10^{-6}$

Table 3.7: Data for Section 3.6.2 with compression tolerance 10^{-6}

N	h	t_{comp}	t_{build}	t_{apply}	mem	res
6400	0.0125	2.61	1.11	0.07	0.36	$4.23 \cdot 10^{-11}$
25600	0.00625	15.81	6.54	0.21	2.01	$7.40 \cdot 10^{-11}$
102400	0.003125	79.86	36.36	1.06	10.38	$2.67 \cdot 10^{-10}$
409600	0.0015625	491.72	206.02	9.55	50.42	$4.59 \cdot 10^{-10}$
1638400	0.00078125	3289.37	1109.28	25.72	234.48	$1.02 \cdot 10^{-8}$

Table 3.8: Data for Section 3.6.2 with compression tolerance 10^{-9}

N	h	t_{comp}	t_{build}	t_{apply}	mem	res
6400	0.0125	2.72	1.23	0.06	0.39	$3.28 \cdot 10^{-14}$
25600	0.00625	16.34	7.20	0.21	2.15	$1.03 \cdot 10^{-13}$
102400	0.003125	82.90	40.68	1.16	10.95	$6.29 \cdot 10^{-13}$
409600	0.0015625	515.89	227.17	5.08	53.34	$3.36 \cdot 10^{-12}$
1638400	0.00078125	3495.09	1304.42	48.21	252.23	$1.02 \cdot 10^{-11}$

Table 3.9: Data for Section 3.6.2 with compression tolerance 10^{-12}

3.6.3 Preconditioned iterative solution for a cavity

The examples in Section 3.6.1 and Section 3.6.2 show that the cost of the direct solver grows substantially as the compression tolerance is tightened. This can be problematic when high-accuracy solutions are required for large-scale problems. As discussed in Section 3.5, this issue can sometimes be circumvented by using the direct solver as a preconditioner. We now apply the approach advocated in Section 3.5 with the 4^{th}-order quadrature corrections used for the preconditioner, GMRES as the iterative solver with convergence tolerance set to 10^{-10}, and compression tolerance 10^{-4} for scattering against the same cavity as in Section 3.6.2. We use the same plane wave (3.99), but this time we increase the wavenumber and keep the number of points per

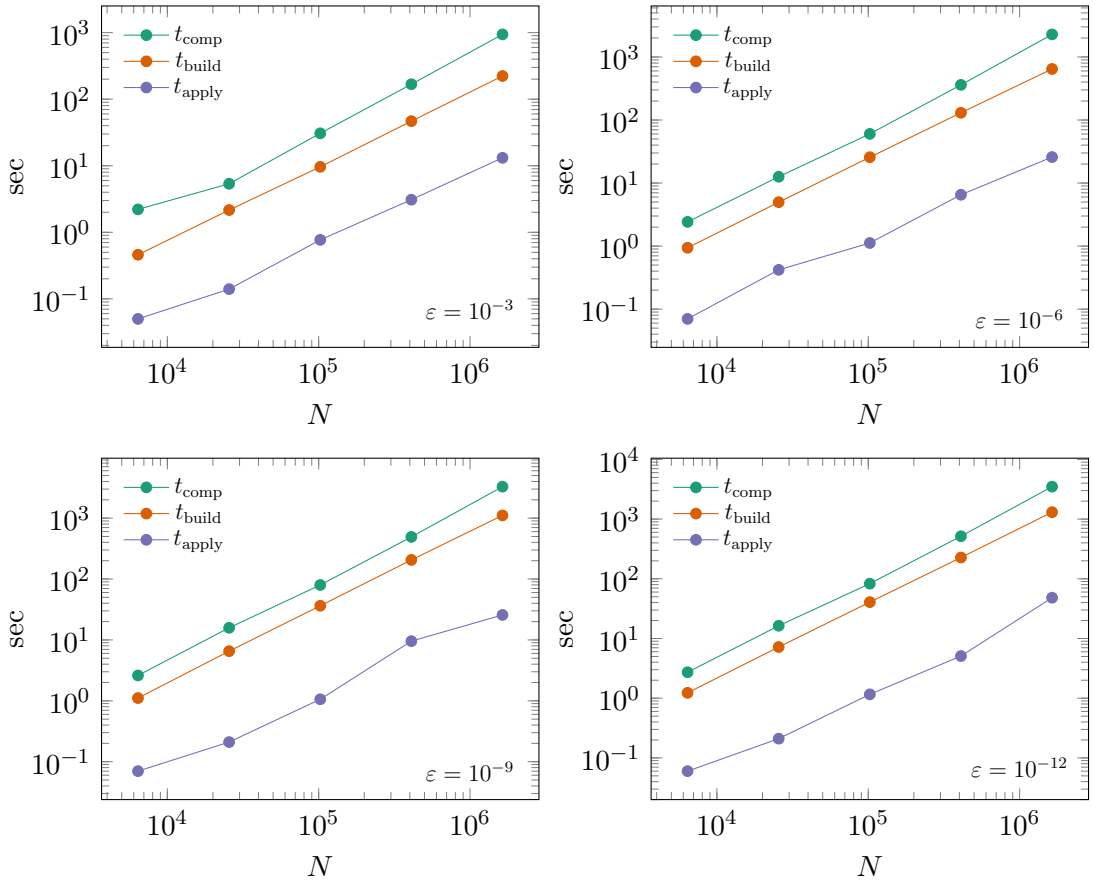


Figure 3.12: Scaling data for Section 3.6.2

wavelength fixed at 10. We consider problems ranging from $\kappa = 50.27$ all the way up to $\kappa = 3216.99$, which corresponds to a computational domain approximately 512 wavelengths across. In addition to the relevant quantities defined in Section 3.6.1, we also record the quantities:

t_{gmres}	Total time spent by GMRES
iter	The number of GMRES iterations required to reach a relative residual with norm below 10^{-10}

This data is presented in Table 3.10, and Fig. 3.13 plots the real part of the total field when $\kappa = 201.06$ and $\kappa = 804.25$.

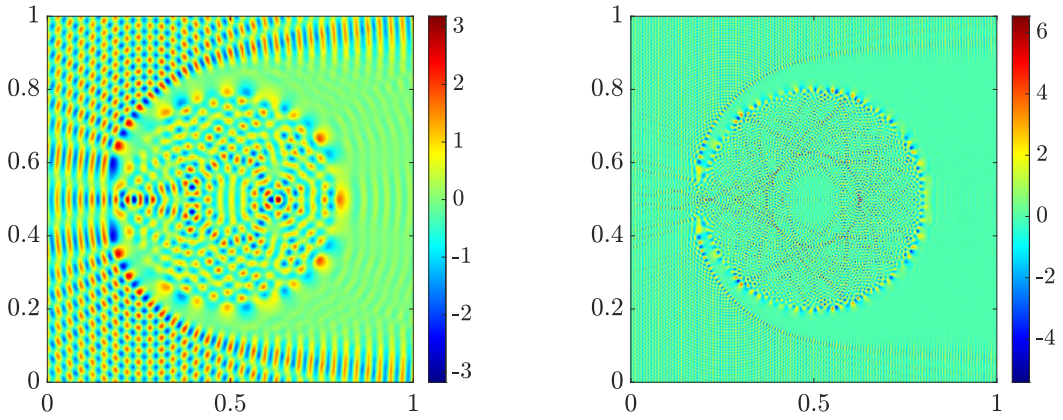


Figure 3.13: Total field that results when a plane wave impinges on a cavity when $\kappa = 201.06$ (a) and $\kappa = 804.25$ (b)

N	κ	t_{comp}	t_{build}	t_{gmres}	mem	iter	res
6400	50.27	0.23	0.24	0.20	0.04	4	$6.97 \cdot 10^{-11}$
25600	100.53	0.65	0.99	0.62	0.21	5	$6.16 \cdot 10^{-12}$
102400	201.06	2.26	4.36	2.49	1.01	6	$1.04 \cdot 10^{-12}$
409600	402.12	14.91	20.06	9.78	4.67	6	$3.23 \cdot 10^{-11}$
1638400	804.25	99.01	91.37	56.13	21.16	9	$8.12 \cdot 10^{-12}$
6553600	1608.50	430.60	398.88	330.91	94.63	13	$3.93 \cdot 10^{-11}$
26214400	3216.99	3102.09	2024.16	2698.53	418.37	22	$3.30 \cdot 10^{-11}$

Table 3.10: Data for Section 3.6.3

The largest problem considered here can be solved to residual error less than 10^{-10} in merely a couple of hours despite having over 26 million degrees of freedom. Without preconditioning, achieving any accuracy for this problem would require thousands of iterations and so would be completely intractable even on a supercomputer.

3.6.4 Preconditioned iterative solution for other geometries

For our final set of numerics, we apply our direct solver to the problems introduced in [61]. Specifically, we study the following problems:

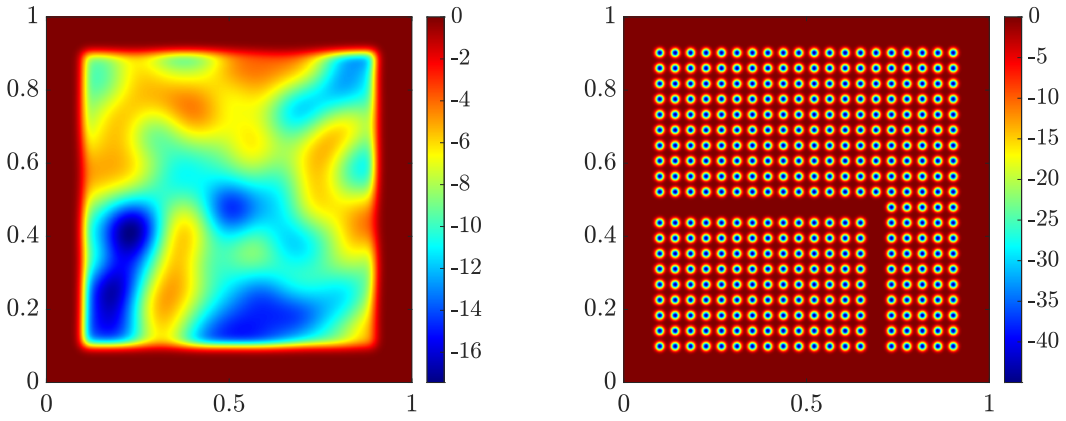


Figure 3.14: (a) Scattering potential for random bumps. (b) Scattering potential for photonic crystal for experiments in Section 3.6.4

Lens	A vertically-graded lens given by (3.26) with $\kappa = 300$ (Fig. 3.3(b)). The incident field is given by the plane wave (3.99). The computational domain is about 47.7 wavelengths wide.
Random bumps	The sum of 200 Gaussian bumps randomly placed in Ω and rolled off to zero in a smooth fashion with $\kappa = 160$ (Fig. 3.14(a)). The incident field is given by a plane wave (3.99). The computational domain is about 25.5 wavelengths wide.
Photonic crystal[3]	A 20 by 20 square grid of Gaussian bumps with a channel removed with $\kappa = 77.1$, which is the first complete band gap (Fig. 3.14(b)). The incident field is given by a plane wave (3.99). The computational domain is about 12.2 wavelengths wide.
Photonic crystal 2	Same geometry and wavenumber as above. The incident field is now given by a plane wave $u_{\text{inc}}(x_1, x_2) = \exp(i\kappa(-x_1/\sqrt{2} + x_2/\sqrt{2} - 1))$.

We again use a preconditioned iterative solver with the same setup as in the previous section. Here, we can also gauge the accuracy in the computed solution by evaluating the computed fields and using the values computed by the spectral solver in [61] as reference values. We report the following quantities:

E_{near}	The absolute error between the real part of the computed solution at the point $(0.75, 0.5)$ and the reference solution given in [61]
E_{far}	The absolute error between the real part of the computed solution at the point $(1.5, 1)$ and the reference solution given in [61]

Tables 3.11, 3.12, 3.13, and 3.14 record the relevant quantities, and the computed total fields are plotted in Fig. 3.15.

[3]The code to compute the photonic crystal was graciously provided by Alex Barnett.

N	h	iter	res	E_{near}	E_{far}
102400	0.003125	51	$6.38 \cdot 10^{-11}$	$1.27 \cdot 10^{-2}$	$2.06 \cdot 10^{-3}$
409600	0.015625	9	$6.18 \cdot 10^{-11}$	$1.05 \cdot 10^{-5}$	$6.77 \cdot 10^{-7}$
1638400	0.00078125	7	$4.86 \cdot 10^{-12}$	$1.14 \cdot 10^{-8}$	$5.22 \cdot 10^{-10}$
6553600	0.000390625	7	$7.64 \cdot 10^{-12}$	$8.29 \cdot 10^{-11}$	$2.81 \cdot 10^{-10}$
26214400	0.0001953125	9	$5.41 \cdot 10^{-12}$	$7.85 \cdot 10^{-11}$	$2.79 \cdot 10^{-10}$

Table 3.11: Data for lens problem in Section 3.6.4

N	h	iter	res	E_{near}	E_{far}
102400	0.003125	100	$3.69 \cdot 10^{-2}$	0.31	$2.53 \cdot 10^{-2}$
409600	0.015625	22	$9.84 \cdot 10^{-11}$	$3.90 \cdot 10^{-4}$	$2.82 \cdot 10^{-4}$
1638400	0.00078125	10	$8.05 \cdot 10^{-12}$	$4.17 \cdot 10^{-7}$	$3.01 \cdot 10^{-7}$
6553600	0.000390625	13	$9.24 \cdot 10^{-11}$	$5.29 \cdot 10^{-10}$	$6.83 \cdot 10^{-10}$
26214400	0.0001953125	23	$3.71 \cdot 10^{-11}$	$1.20 \cdot 10^{-10}$	$9.90 \cdot 10^{-10}$

Table 3.12: Data for random bumps problem in Section 3.6.4

N	h	iter	res	E_{near}	E_{far}
102400	0.003125	24	$6.49 \cdot 10^{-11}$	$1.67 \cdot 10^{-3}$	$1.09 \cdot 10^{-3}$
409600	0.015625	8	$3.70 \cdot 10^{-12}$	$2.96 \cdot 10^{-6}$	$1.47 \cdot 10^{-6}$
1638400	0.00078125	7	$2.30 \cdot 10^{-12}$	$3.68 \cdot 10^{-9}$	$1.29 \cdot 10^{-9}$
6553600	0.000390625	15	$2.24 \cdot 10^{-11}$	$4.81 \cdot 10^{-10}$	$2.94 \cdot 10^{-10}$
26214400	0.0001953125	11	$5.23 \cdot 10^{-11}$	$4.57 \cdot 10^{-10}$	$3.00 \cdot 10^{-10}$

Table 3.13: Data for first photonic crystal problem in Section 3.6.4

N	h	iter	res	E_{near}	E_{far}
102400	0.003125	24	$9.99 \cdot 10^{-11}$	$5.15 \cdot 10^{-3}$	$4.01 \cdot 10^{-3}$
409600	0.015625	8	$2.36 \cdot 10^{-12}$	$7.12 \cdot 10^{-6}$	$5.97 \cdot 10^{-6}$
1638400	0.00078125	7	$1.82 \cdot 10^{-12}$	$7.88 \cdot 10^{-9}$	$1.15 \cdot 10^{-8}$
6553600	0.000390625	14	$8.33 \cdot 10^{-11}$	$1.67 \cdot 10^{-10}$	$5.01 \cdot 10^{-9}$
26214400	0.0001953125	11	$4.75 \cdot 10^{-11}$	$1.27 \cdot 10^{-10}$	$5.00 \cdot 10^{-9}$

Table 3.14: Data for second photonic crystal problem in Section 3.6.4

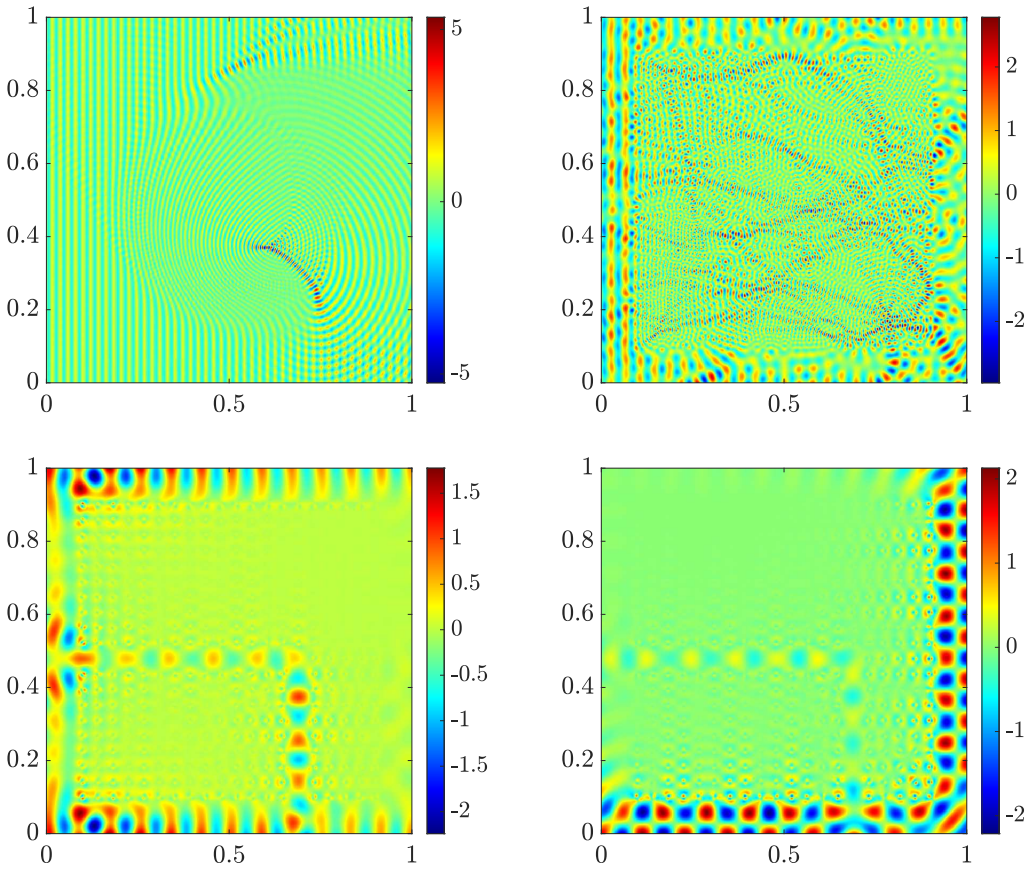


Figure 3.15: The real parts of the total fields for the problems from [61] in Section 3.6.4

Chapter 4

Broadband recursive skeletonization[1]

4.1 Introduction

In this chapter we return to homogeneous Helmholtz equations, which we will consider in both two and three dimensions. We will take an approach akin to that of Chapter 3 and reformulate our problem as an integral equation. For a time-harmonic, exterior scattering problem with wavenumber $\kappa > 0$ and scatterer $\Omega \subset \mathbb{R}^d$ with boundary Γ for $d = 2$ or 3, we can typically represent the scattered field using an ansatz of the form

$$u(\boldsymbol{x}) = \int_\Gamma \left(\frac{\partial G_\kappa}{\partial n_{\boldsymbol{y}}}(\boldsymbol{x}, \boldsymbol{y}) - i\eta G_\kappa(\boldsymbol{x}, \boldsymbol{y}) \right) \sigma(\boldsymbol{y})\, ds(\boldsymbol{y}) \tag{4.1}$$

for some $\eta \geq 0$ and $\boldsymbol{x} \in \mathbb{R}^d \backslash \overline{\Omega}$ where

$$G_\kappa(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \dfrac{i}{4} H_0(\kappa \|\boldsymbol{x} - \boldsymbol{y}\|), & d = 2, \\ \dfrac{1}{4\pi} \dfrac{e^{i\kappa\|\boldsymbol{x}-\boldsymbol{y}\|}}{\|\boldsymbol{x} - \boldsymbol{y}\|}, & d = 3 \end{cases} \tag{4.2}$$

is the free-space fundamental solution of the Helmholtz equation. The density function σ in (4.1) is then chosen so that u satisfies the given boundary conditions. For example, for a sound-soft exterior problem with a given incident field u_{inc}, σ should be chosen to satisfy the integral equation

$$\frac{1}{2}\sigma(\boldsymbol{x}) + \int_\Gamma \left(\frac{\partial G_\kappa}{\partial n_{\boldsymbol{y}}}(\boldsymbol{x}, \boldsymbol{y}) - i\eta G_\kappa(\boldsymbol{x}, \boldsymbol{y}) \right) \sigma(\boldsymbol{y})\, ds(\boldsymbol{y}) = -u_{\text{inc}}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma. \tag{4.3}$$

[1]The content of this chapter will be presented in a forthcoming preprint by AG and his collaborator PGM. The key idea in this work of building a universal basis for integral operators across a range of frequencies was proposed by AG. AG developed the theory, algorithms, and ran the numerical experiments with advice and insights from PGM.

The equation (4.3) is derived from the so-called jump relations for layer potentials [44, Theorem 3.1] and is known in the literature as the combined field integral equation. It is known to be well-posed as an integral equation on $L^2(\Gamma)$ or $C(\Gamma)$ for any $\kappa > 0$ and $\eta > 0$ [44, Theorem 3.11]. For interior problems the ansatz (4.1) where now $\boldsymbol{x} \in \Omega$ may still be used, and the analog of (4.3) is given by

$$-\frac{1}{2}\sigma(\boldsymbol{x}) + \int_\Gamma \left(\frac{\partial G_\kappa}{\partial n_{\boldsymbol{y}}}(\boldsymbol{x}, \boldsymbol{y}) - i\eta G_\kappa(\boldsymbol{x}, \boldsymbol{y}) \right) \sigma(\boldsymbol{y})\, ds(\boldsymbol{y}) = -u_{\text{inc}}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma; \quad (4.4)$$

there exist similar integral equations for sound-hard problems.

As in the methods of the previous two chapters, the choice of ansatz guarantees that the computed scattered field will solve the Helmholtz equation, satisfy the Sommerfeld radiation condition, and can be discretized without introducing dispersion errors. Since we are in the homogeneous setting, the integrals in (4.1), (4.3), and (4.4) are over the boundary of the scatterer, which leads to a reduction in the dimension of the region that must be discretized. As such, fewer degrees of freedom are required, and it is unnecessary to mesh volumes or unbounded regions. After scaling by a constant, (4.3) and (4.4) are Fredholm integral equations of the second kind on $L^2(\Gamma)$ or $C(\Gamma)$ [103, Theorem 2.29] and can thus be efficiently discretized without introducing spurious ill-conditioning, which is particularly desirable in near-resonant or high-frequency problems. Also, the square linear systems obtained from discretizing boundary integral equations tend to be more amenable to existing techniques for fast numerical linear algebra than methods based on least squares, so working with the former is preferable for large-scale problems.

As discussed in Section 3.1, most methods of discretizing integral equations lead to dense coefficient matrices that are amenable to inversion using $\mathcal{H}$-matrices. While there are a variety of inversion algorithms using different types of $\mathcal{H}$-matrix formats, all of them can be conceptually divided into the same three-stage process as the solver introduced in Sections 3.3 and 3.4. To recapitulate, each of these algorithms first executes a *compression stage* where an approximation in a rank-structured format to the coefficient matrix is sought. This consists of computing approximate, low-rank factorizations of various off-diagonal blocks of the coefficient matrix. Then there is a *build stage* where the rank-structured approximation is inverted. Naturally, the inverse is also given in a rank-structured format. This is exploited in the *apply stage* where the inverse is applied to a vector. The exact computational complexity of each of the three stages depends on the specifics of both the algorithm and the problem.

We will specifically focus on direct solvers that use *recursive skeletonization* (RS), which is when a Nyström discretization is coupled with the interpolative decomposition

(ID) for the low-rank factorizations to accelerate the compression stage (e.g., [28, 38, 45, 62, 65, 73, 88–90, 101, 112, 113, 144]). Due to developments in recent years, there now exist practical direct solvers using RS for which all three stages are at least quasilinear in the number of degrees of freedom in both 2D (e.g., [62, 90, 113]) and 3D (e.g., [5, 45, 89, 120]) for a broad range of applications. Even for these solvers, the constants in the complexity of each stage tend to vary drastically, resulting in very different timings in practice. While the specifics again depend on the algorithm and problem, it is fairly typical, at least for weakly-admissible formulations (see Remark 3.5), for the compression stage to be a factor of 5–20 times slower than the build stage, which is in turn 10–100 times slower than the apply stage. In applications such as preconditioning, systems must be solved with the same coefficient matrix and different right-hand sides, so this breakdown of costs is rather convenient since the compression and build stages must be executed only once; the apply stage is then repeated many times.

However, in other environments, such as when evaluating radar cross sections, eigenvalue problems [176], and Fourier transform-based solvers for the wave equation [8], equations of the form (4.3) and (4.4) must be solved at many different wavenumbers in some interval $[\kappa_{\min}, \kappa_{\max}]$ where $\kappa_{\max} > \kappa_{\min} > 0$. There is a small body of literature which computes interpolants, with respect to the wavenumber, for the entries of the coefficient matrices arising in boundary element discretizations to accelerate adaptive cross approximation in this environment [99, 147, 148, 160]. Something similar could perhaps be incorporated in RS, but even so this would only yield a minor acceleration. That aside, the standard prescription for applying RS-based direct solvers in this setting is to simply repeat all three stages at each wavenumber of interest. Since this requires repeating the costly compression stage many times, this can be rather expensive. In this chapter we introduce the *broadband recursive skeletonization* (BRS) method for accelerating RS in this regime. The key observation here is that instead of computing low-rank factorizations of the relevant off-diagonal blocks from scratch at each wavenumber, one can instead do a one-time computation to find a *universal basis* for the column or row space of each block that is valid for all $\kappa \in [\kappa_{\min}, \kappa_{\max}]$. Then for each wavenumber, all that remains to be done in the compression stage is to project each block onto its respective universal basis. This can be done efficiently since the ID is used. The trade-offs here are that the number of degrees of freedom in the discretization must be kept constant across the wavenumber interval and the ranks in the resulting rank-structured matrices are suboptimal. Since these solvers usually scale linearly or almost linearly in the number of degrees of freedom, the former issue

is generally of little concern. The latter issue, on the other hand, is more delicate since the complexities typically scale at least quadratically with the ranks.

Fortunately, it turns out that for a large class of geometries, the ranks produced when the universal bases are used are only slightly higher than the optimal ranks at the highest wavenumber $\kappa_{\max}$. This can be attributed to the bandlimited nature of solutions to the Helmholtz equation. Indeed, in RS finding a basis for the column space of a given off-diagonal block in the rank-structured format may be interpreted as finding a basis for the space of all incoming fields on a set of targets located on one piece of the boundary of the scatterer generated by sources located elsewhere. A universal basis for the column space then amounts to a basis for incoming fields with the same targets and sources but now for all wavenumbers $\kappa \in [\kappa_{\min}, \kappa_{\max}]$. Since all such fields are oscillatory functions restricted to a curve in 2D or surface in 3D, they are approximately bandlimited, which we can exploit. The ultimate result is that we can often greatly reduce the cost of the compression stage at each wavenumber, while only slightly increasing the costs of the other stages after a one-time computation of the universal bases. In several different regimes of interest, we see a speedup of a factor of about 2–10 in the per wavenumber cost when BRS is used.

We now outline what follows. In Section 4.2 we review some preliminaries necessary to develop our theory and algorithms. Specifically, we discuss the discretization of equations of the form (4.3) and (4.4), as well as some basic facts about RS and prolate spheroidal wave functions. Section 4.3 lays out the key theoretical observations that we exploit in this work. This is followed by Section 4.4 where we turn our attention to the efficient computation of the universal bases. This chapter ends on Section 4.5, which illustrates our method in several different regimes using numerical experiments.

4.2 Preliminaries

4.2.1 Discretization

In Section 3.2.2 we discussed the discretization of the Lippmann–Schwinger integral equation. In this section we review the analogous approach for boundary integral operators and equations in a slightly more abstract setting. We start by considering two curves or surfaces Γ_s and Γ_t, and a generic integral operator $\mathcal{K} : C(\Gamma_s) \to C(\Gamma_t)$ of the form

$$[\mathcal{K}\sigma](\boldsymbol{x}) = \int_{\Gamma_s} K(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y})\, ds(\boldsymbol{y}), \quad \boldsymbol{x} \in \Gamma_t \tag{4.5}$$

where $\sigma : \Gamma_s \to \mathbb{C}$ and $K : \Gamma_t \times \Gamma_s \to \mathbb{C}$. Assuming temporarily that K is smooth over $\Gamma_t \times \Gamma_s$, it is natural to discretize the operator using a quadrature rule $\{(w_j, \boldsymbol{y}_j)\}_{j=1}^N$ for which

$$\int_{\Gamma_s} K(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y})\, ds(\boldsymbol{y}) \approx \sum_{j=1}^{N} w_j K(\boldsymbol{x}, \boldsymbol{y}_j)\sigma(\boldsymbol{y}_j), \quad \boldsymbol{x} \in \Gamma \tag{4.6}$$

for all sufficiently smooth σ. Now, suppose $\{(w_i', \boldsymbol{x}_i)\}_{i=1}^M$ is a quadrature rule for which

$$\int_{\Gamma_t} u(\boldsymbol{x})\, ds(\boldsymbol{x}) \approx \sum_{i=1}^{M} w_i' u(\boldsymbol{x}_i) \tag{4.7}$$

for sufficiently smooth $u : \Gamma_t \to \mathbb{C}$. Then we have

$$\sqrt{w_i'}[\mathcal{K}\sigma](\boldsymbol{x}_i) \approx [\mathbf{K}\boldsymbol{\sigma}](i) \tag{4.8}$$

where $\boldsymbol{\sigma} \in \mathbb{C}^N$ and $\mathbf{K} \in \mathbb{C}^{M\times N}$ have entries given by

$$\boldsymbol{\sigma}(j) = \sqrt{w_j}\sigma(\boldsymbol{y}_j), \text{ and } \mathbf{K}(i,j) = \sqrt{w_i'}\sqrt{w_j}K(\boldsymbol{x}_i, \boldsymbol{y}_j) \tag{4.9}$$

for $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$. The entries in (4.9) are scaled by the square roots of the quadrature weights so that the discrete inner products approximate the L^2-inner products. This can be important for conditioning [28].

If $\Gamma_s = \Gamma_t$, we can apply the above discretization scheme to integral equations of the form

$$\lambda\sigma(\boldsymbol{x}) + \mathcal{K}[\sigma](\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma_t \tag{4.10}$$

for some $\lambda \in \mathbb{C}$ and sufficiently smooth $f : \Gamma_t \to \mathbb{C}$ by imposing a collocation condition at the points $\{\boldsymbol{x}_i\}_{i=1}^N$, which should be chosen to be appropriate interpolation points for both f and σ. This yields the algebraic equations

$$\lambda\sigma_i + \sqrt{w_i'}\sum_{j=1}^{N}\sqrt{w_j}K(\boldsymbol{x}_i, \boldsymbol{y}_j)\sigma_j' = \sqrt{w_i'}f(\boldsymbol{x}_i), \quad i = 1, 2, \ldots, N \tag{4.11}$$

where the hope is that $\sigma_i \approx \sqrt{w_i'}\sigma(\boldsymbol{x}_i)$ and $\sigma_j' \approx \sqrt{w_j}\sigma(\boldsymbol{y}_j)$ for $i, j = 1, 2, \ldots, N$ and sufficiently large N. While (4.11) has $2N$ unknowns a priori, we have chosen the collocation points so that $\{\sigma_j'\}_{j=1}^N$ may be interpolated from $\{\sigma_i\}_{i=1}^N$.

We can simplify this, as well as gain a few further niceties, by using a *Nyström discretization*, which refers to the case when the collocation points and quadrature points are chosen to be the same. In our above notation this corresponds to taking $\boldsymbol{y}_i = \boldsymbol{x}_i$ for $i = 1, 2, \ldots, N$. We will restrict ourselves to this case in the sequel, and with this assumption, the equations in (4.11) can by expressed as the linear system

$$\mathbf{G}\boldsymbol{\sigma} = \mathbf{f} \tag{4.12}$$

where $\mathbf{f}(i) = \sqrt{w_i} f(\boldsymbol{x}_i)$, $\boldsymbol{\sigma}(i) = \sigma_i$, and

$$\mathbf{G}(i,j) = \begin{cases} \lambda + w_i K(\boldsymbol{x}_i, \boldsymbol{x}_i), & i = j, \\ \sqrt{w_i}\sqrt{w_j} K(\boldsymbol{x}_i, \boldsymbol{x}_j), & i \neq j \end{cases} \tag{4.13}$$

for $i, j = 1, 2, \ldots, N$.

So far, we have restricted ourselves to the case where the kernel function is completely smooth. When K has a singularity when $\boldsymbol{x} = \boldsymbol{y}$, the more common case in applications, the assumption that we can obtain a quadrature rule such that (4.6) holds is no longer realistic. Instead, we must use a different quadrature rule for each collocation node. This can affect the rank structure of the coefficient matrix since it will no longer simply be a diagonal scaling of a kernel matrix. Fortunately, there exist a plethora of quadrature rules that result in only $\mathcal{O}(N)$ entries in $\mathbf{G}$ deviating from (4.13), known as FMM-compatible quadrature rules [84]. These quadrature rules typically work by making local corrections near the singularity. For example, restricting ourselves to the ith row of $\mathbf{G}$, which enforces the collocation condition at the node $\boldsymbol{x}_i$, these rules would correct the entries $\mathbf{G}(i,j)$ for j such that $\boldsymbol{x}_j$ is in a stencil around $\boldsymbol{x}_i$. The size and shape of the stencil depends on the type and order of the rule, but in every case only $\mathcal{O}(N)$ entries need to be modified and so most of the rank structure is preserved and fast direct solver techniques can be used. Examples of FMM-compatible quadrature rules include Kapur–Rokhlin [95], Alpert [2], Kolm–Rokhlin [100], Xiao–Gimbutas [173], Vioreanu–Rokhlin [159], and zeta corrections [169, 170]. We refer the reader to the survey paper [84] for more discussion on quadrature rules for singular integrals in 2D.

It is an established result that for a well-posed second kind integral equation solutions produced through a Nyström discretization will converge to the true solution at the same rate as the quadrature rule under some mild assumptions on the domain, the kernel, and the quadrature [9, 11, 103].

4.2.2 Recursive skeletonization

The core ideas in recursive skeletonization (RS) were actually introduced in detail in Section 3.3 in the context of a 2D area integral operator. Of course, instead of hierarchically subdividing Ω, we can now subdivide just its boundary Γ. In this case the root node of the tree will correspond to the entire boundary $\Gamma_1 = \Gamma$ and the associated index vector $\mathcal{I}_1$ will contain the indices of all of the degrees of freedom in the discretization. On the next level Γ_1 will be subdivided into two halves Γ_2 and Γ_3, with respective index vectors $\mathcal{I}_2$ and $\mathcal{I}_3$ which contain the indices of the degrees of

freedom that lie in the respective region. Again, this subdivision will be repeated until the nodes at the lowest level in the tree (the leaves) contain sufficiently few points.

After the tree structure has been created, we start at the bottom level of the tree and compress the interactions between each leaf and all other leaves. Since the ID is used, this amounts to choosing representative degrees of freedom for each leaf node which we call *skeleton points*, onto which the node's outgoing field can be projected and from which its incoming field can be interpolated. Again, the method of proxy sources is used. After this we move one level up the tree to the level of the parents of the leaves and regroup the previously obtained skeleton points according to sibling pairs. The interactions between these new groups are then compressed, again using the ID to obtain new skeleton points. This process of regrouping and recompression is then repeated all the way up the tree. In the language of the fast multipole method (FMM), this can be viewed as computing translation operators tailored to the specific geometry (cf. [72,74]). The projection and interpolation matrices in the ID correspond to the multipole-to-multipole and local-to-local translation operators, respectively, and submatrices of the coefficient matrix correspond to the multipole-to-local translation operators. Fig. 4.1 sketches the idea of RS for a contour in 2D. For a more thorough treatment of RS, we refer the reader to Section 3.3 and the references therein.

The discussions of RS in Section 3.3 and the above paragraphs were based on hierarchically block separable (HBS) matrices, which were introduced in Section 3.2.4. As mentioned in Remark 3.5, the HBS format is a weakly-admissible format since it allows us to compress blocks corresponding to interactions between patches that are adjacent in space. In contrast are strongly-admissible formats, which only compress the interactions between patches of space that are sufficiently separated as determined by a so-called admissibility condition. The strongly-admissible analog of HBS matrices are usually just referred to as $\mathcal{H}^2$-matrices (although, sometimes in the literature HBS matrices are referred to as weakly-admissible $\mathcal{H}^2$-matrices). In this case it is typically more natural to use a quad-tree instead of a binary tree for the hierarchical structure, and the inversion algorithms are more complicated. With these changes the extension of RS to strongly-admissible formulations is relatively straightforward and is spelled out in detail in [120], for example. The ideas in this chapter apply to both weakly- and strongly-admissible formulations.

4.2.3 Prolate spheroidal wave functions

Important to our theory are the *prolate spheroidal wave functions of order zero* (PSWFs) and their analog in higher dimensions, the *generalized prolate spheroidal*

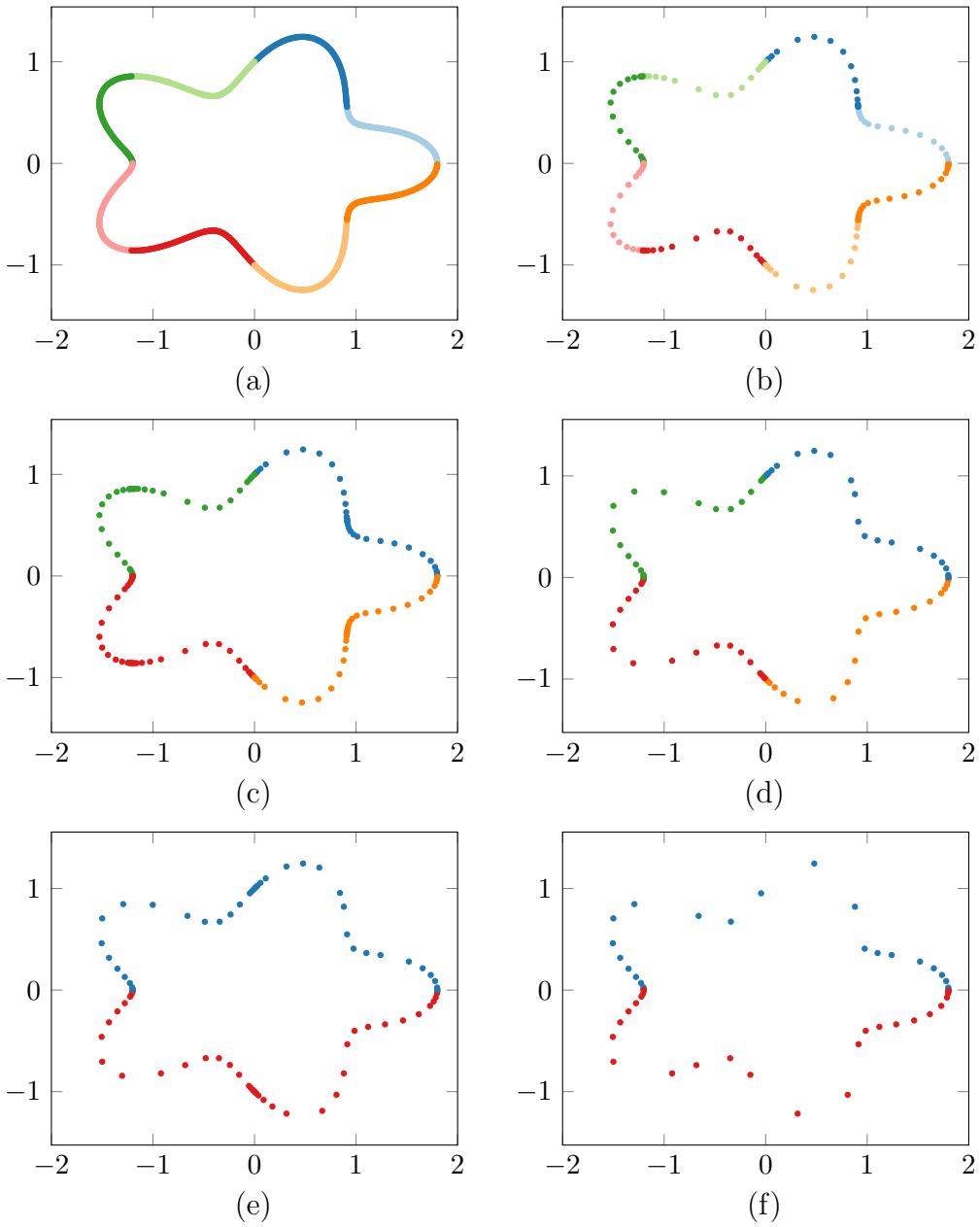


Figure 4.1: Illustration of four-level recursive skeletonization for a contour in 2D. Each piece of the contour corresponding to a different leaf is shown in a different color in (a). After the interactions between the leaves are compressed, the skeleton points that remain are shown in (b). These skeleton points are then grouped by sibling pairs to introduce new low-rank structure in (c). The interactions between these new groups are again compressed to obtain the skeleton points in (d). This procedure is then repeated in (e) and (f).

functions (GPSFs). The PSWFs arise in the separation of variables solution of the Helmholtz equation in prolate spheroidal coordinates. Remarkably, these functions also play a fundamental role in signal processing and related applications.

This can be seen by fixing a parameter $c > 0$, called the *bandlimit*, and considering the truncated Fourier transform operator $\mathcal{F}_c : L^2([-1,1]) \to L^2([-1,1])$ defined by

$$[\mathcal{F}_c\sigma](x) = \int_{-1}^{1} e^{ic\xi x}\sigma(\xi)\,d\xi, \quad x \in [-1,1] \tag{4.14}$$

for $\sigma \in L^2([-1,1])$. It turns out that the eigenfunctions of $\mathcal{F}_c$, which we will denote by $\{\psi_n^c\}_{n=0}^{\infty}$, are precisely PSWFs. Specifically, these PSWFs satisfy

$$\int_{-1}^{1} e^{ic\xi x}\psi_n^c(\xi)\,d\xi = \lambda_n^c\psi_n^c(x), \quad x \in [-1,1] \tag{4.15}$$

for all $n = 0, 1, \ldots$. It is known that all of the eigenvalues $\{\lambda_n^c\}_{n=0}^{\infty}$ have multiplicity 1 and are bounded above in magnitude, and that the $\{\psi_n^c\}_{n=0}^{\infty}$ are real-valued, orthonormal, and complete in $L^2([-1,1])$ [140]. We use the standard convention of ordering the PSWFs so that $|\lambda_0^c| > |\lambda_1^c| > \ldots > 0$. Alternatively, the PSWFs can also be characterized as eigenfunctions of the bandlimiting and space-limiting operator on $L^2([-1,1])$ given by

$$\mathcal{Q}_c = \frac{c}{2\pi}\mathcal{F}_c^*\mathcal{F}_c, \tag{4.16}$$

which is an integral operator of the form

$$[\mathcal{Q}_c\sigma](x) = \frac{1}{\pi}\int_{-1}^{1} \frac{\sin(c(x-\xi))}{x-\xi}\sigma(\xi)\,d\xi \tag{4.17}$$

for $\sigma \in L^2([-1,1])$.

Our interest in PSWFs is motivated by the fact that they are a convenient basis for the approximation of functions whose Fourier transforms are compactly supported. We say that a function $f : [-1,1] \to \mathbb{C}$ is *bandlimited* with bandlimit $c > 0$ if it can be written

$$f(x) = \int_{-1}^{1} e^{ic\xi x}\,d\mu(\xi) \tag{4.18}$$

for some complex-valued measure μ with $\int_{-1}^{1} d|\mu|(\xi) < \infty$. The following result combines Lemma 3.5 and Remark 2.4 in [137], and states that the error in approximating a bandlimited function with a truncated expansion in the PSWFs depends on the rate of decay of the modulus of the eigenvalues $\{\lambda_n^c\}_{n=0}^{\infty}$.

Lemma 4.1 (Shkolnisky, Tygert, and Rokhlin 2006)**.** *Let f be a bandlimited function of the form (4.18). Then for any $N \in \mathbb{N}$, the following relation in the above notation holds*

$$\left| f(x) - \sum_{n=0}^{N} f_n \psi_n^c(x) \right| \leq 4 \left(\sum_{n=N+1}^{\infty} n|\lambda_n^c| \right) \int_{-1}^{1} d|\mu|(\xi) \tag{4.19}$$

for all $x \in [-1, 1]$ where

$$f_n = \int_{-1}^{1} f(x) \psi_n^c(x)\, dx. \tag{4.20}$$

While there are a myriad of results on the behavior of the eigenvalues of the operator (4.14) (for example, see Chapter 5 in [126]), the vast majority of these are asymptotic, in terms of complicated special functions that are not easily interpretable, or too loose to be of practical value for us here. As such, we instead state a conjecture made by Osipov, Rokhlin, and Xiao based on extensive numerical evidence, which to the best of our knowledge, has yet to be proven. The following is Conjecture 1 in [126].

Conjecture 4.2 (Osipov, Xiao, and Rokhlin 2013)**.** *Assume that $c > 1$ and let $\varepsilon \in (0, 1)$ be given. Then for any $n \in \mathbb{N}$ such that*

$$n > \frac{2c}{\pi} + 10 + \frac{2}{\pi^2} (\log c) \log \frac{1}{\varepsilon}, \tag{4.21}$$

we have

$$|\lambda_n^c| < \varepsilon. \tag{4.22}$$

Since $[-1, 1]$ is c/π wavelengths of the plane wave e^{icx} long, Conjecture 4.2 implies that the error in the right-hand side of (4.19) will exponentially decay when there are slightly more than two terms per wavelength in the expansion. This is of course closely related to the Shannon–Nyquist sampling theorem in signal processing. For more on the PSWFs, we refer the readers to the textbook [126] and the papers [26, 106, 107, 137, 139, 140, 173].

As aforementioned, the PSWFs generalize to GPSFs in higher dimensions. While in principle the interval $[-1, 1]$ in (4.14) may be replaced by any set in $\mathbb{R}^d$ (see [138]), the majority of the literature focuses on the case of a disk. Specifically, letting D denote

the unit disk in $\mathbb{R}^d$, the GPSFs are most frequently defined as the eigenfunctions of the operator $\mathcal{F}_c : L^2(D) \to L^2(D)$ given by

$$[\mathcal{F}_c\sigma](\boldsymbol{x}) = \int_D e^{ic\boldsymbol{x}\cdot\boldsymbol{\xi}}\sigma(\boldsymbol{\xi})\,d\boldsymbol{\xi}. \tag{4.23}$$

The analog of $\mathcal{Q}_c$, which is now an operator on $L^2(D)$, is now given by the formula

$$[\mathcal{Q}_c\sigma](\boldsymbol{x}) = \left(\frac{c}{2\pi}\right)^{\frac{d}{2}} \int_D \frac{J_{\frac{d}{2}}(c\|\boldsymbol{x}-\boldsymbol{\xi}\|)}{\|\boldsymbol{x}-\boldsymbol{\xi}\|^{\frac{d}{2}}}\sigma(\boldsymbol{\xi})\,d\boldsymbol{\xi}. \tag{4.24}$$

A derivation of this can be found in [75, Section 3.9.1]. It should come as no surprise that less seems to be known about the spectral properties of (4.23) and (4.24) when compared to the 1D case. It is known, however, that when $c \to \infty$ there are to leading order

$$\frac{c^d}{2^d\Gamma\left(\frac{d}{2}+1\right)^2} \tag{4.25}$$

eigenvalues of (4.24) of modulus order 1 and the rest exponentially decay [75, Theorem 3.6.5]. More results on the GPSFs can be found in the dissertation [75] and the papers [76, 108, 136, 138].

4.3 Theoretical foundations

4.3.1 Rank behavior at a single frequency

It is instructive to first review what is known for a single frequency. We will temporarily restrict ourselves to the two-dimensional case and consider various smooth source curves $\Gamma_s \subset \mathbb{R}^2$ and target curves $\Gamma_t \subset \mathbb{R}^2$. We assume that Γ_s and Γ_t are disjoint except possibly at a finite number of points. For simplicity we will limit our discussion to the case of the single layer operator $\mathcal{S}_\kappa : L^2(\Gamma_s) \to L^2(\Gamma_t)$ given by

$$[\mathcal{S}_\kappa\sigma](\boldsymbol{x}) = \frac{i}{4}\int_{\Gamma_s} H_0(\kappa\|\boldsymbol{x}-\boldsymbol{y}\|)\sigma(\boldsymbol{y})\,ds(\boldsymbol{y}), \quad \boldsymbol{x} \in \Gamma_t, \tag{4.26}$$

but all concepts that will be discussed immediately carry over to the double and combined layer operators as well. Since (4.26) is an integral operator with a weakly-singular kernel, it is compact and therefore possesses a singular value expansion.

It is well-known in the scattering theory literature that there will be $\mathcal{O}(\kappa)$ leading singular values of (4.26) of all roughly the same order, which correspond to modes necessary to resolve the oscillatory behavior of elements in the range of $\mathcal{S}_\kappa$. After the oscillations are resolved, the remaining singular values decay exponentially. Pictorially,

we will observe a "knee", when the singular values of $\mathcal{S}_\kappa$ are plotted, where this transition occurs. See Fig. 4.2(b), for example.

The location of the knee depends on the geometry and configuration of the source and target curves, but there are two useful heuristics that provide rough upper bounds. The first is derived from the separability of the Hankel function under the assumption that Γ_s and Γ_t are sufficiently separated and states that the location of the knee will depend on the "angle of interaction" between the sources and targets [39, 55–57, 131]. If this angle is $\theta \in (0, 2\pi]$, then it follows from these arguments that the knee will occur around

$$\frac{\theta\kappa \min\{\operatorname{diam}\Gamma_s, \operatorname{diam}\Gamma_t\}}{2\pi}. \tag{4.27}$$

Note that when $\theta = 2\pi$, this simplifies to the standard multipole estimate, which states that if Γ_s and Γ_t are well-separated, then replacing the Hankel function in (4.26) with a truncated multipole expansion yields a low-rank approximation to $\mathcal{S}_\kappa$ that converges exponentially after there are approximately

$$\kappa \min\{\operatorname{diam}\Gamma_s, \operatorname{diam}\Gamma_t\} \tag{4.28}$$

terms in the expansion [39, 48, 125, 131, 141]. This estimate can be rigorously extended to many configurations where Γ_s and Γ_t are not well-separated using an exponentially graded subdivision argument (see the proof of Lemma 2.1 in [71], for example) and is empirically observed to hold even in situations where such arguments do not apply.

The other heuristic stems from the fact that scattered fields are approximately spatially bandlimited functions, and therefore by roughly the same reasoning as the Shannon–Nyquist sampling theorem, we expect the knee to occur around

$$\frac{\kappa \min\{|\Gamma_s|, |\Gamma_t|\}}{\pi} \tag{4.29}$$

again assuming that Γ_s and Γ_t are sufficiently separated [31, 32, 39, 149, 165]. Note that (4.29) corresponds to two times the minimum length of Γ_s and Γ_t measured in wavelengths. We are not aware of any results that rigorously establish this that are analogous to those cited for the earlier estimate; nonetheless, extensive numerical evidence motivates the validity of this heuristic, and we will return to this notion of bandlimit in slightly more detail later.

Which of these two heuristics is more relevant for a specific geometry depends on how the diameters of Γ_s and Γ_t compare to their arc lengths. Clearly, the utility of the second heuristic deteriorates in cases where Γ_s and Γ_t are approximately space filling, since then the first heuristic will provide a tighter bound. In most settings where

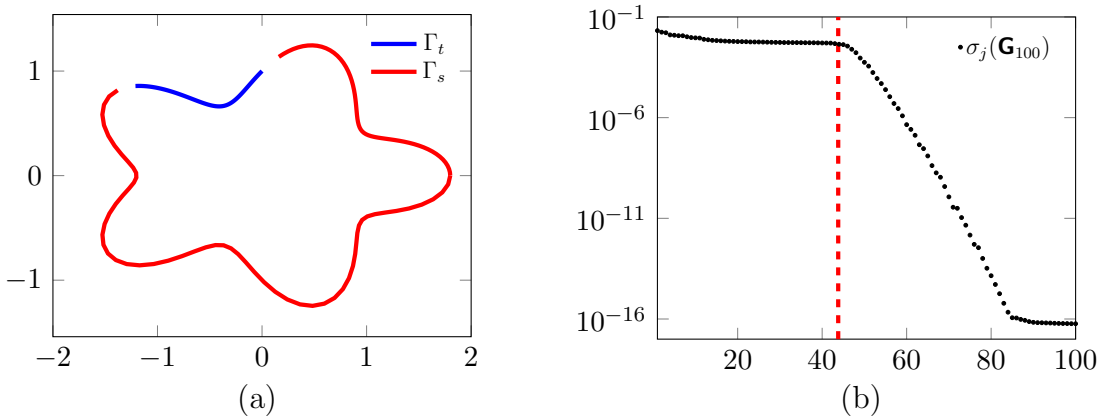


Figure 4.2: (a) Geometry of the model problem considered in Sections 4.3.1, 4.3.2, and 4.3.3. (b) First 100 singular values of $\mathbf{G}_{100}$ computed in double precision arithmetic. The red, dashed line demarcates the values $\kappa|\Gamma_t|/\pi \approx 43.80$.

direct solvers are of interest, this is not the case. It is also worth pointing out that these two heuristics are intimately related. In fact, the first heuristic can be derived by applying the second heuristic to a circular arc separating Γ_s and Γ_t, instead of the curves themselves (see, e.g., [39, Section 2.2.8]).

We illustrate the above ideas by considering a specific geometry where Γ_s and Γ_t consist of pieces of a smooth, five-pointed star contour as illustrated in Fig. 4.2(a). Note that in this case, $|\Gamma_s| \approx 8.50$ and $|\Gamma_t| \approx 1.38$. We let $\mathbf{G}_\kappa \in \mathbb{C}^{m\times n}$ denote a discretization of $\mathcal{S}_\kappa$ in (4.26) in the manner described in Section 4.2.1 and plot the first 100 singular values of $\mathbf{G}_{100}$ in Fig. 4.2(b) computed in double precision. The second heuristic is tighter than the first in this case and estimates that the knee will occur approximately at $\kappa|\Gamma_t|/\pi$. When $\kappa = 100$, in which case Γ_t is about 21.90 wavelengths long, the heuristic suggests that the $\sigma_j(\mathbf{G}_{100})$ will start to exponentially decay when

$$j \approx \frac{100|\Gamma_t|}{\pi} \approx 43.80, \tag{4.30}$$

which is the value marked by the red, dashed line in Fig. 4.2(b). It is clear that the computed singular values closely match this predicted behavior. Specifically, there are about 46 singular values that are all of order 10^{-2}, and then the remaining singular values exponentially decay until machine precision is reached, at which point stagnation occurs.

4.3.2 Introduction to universal skeletons

For our direct solvers we will need to compute low-rank approximations to matrices like $\mathbf{G}_\kappa$ to some given tolerance $\varepsilon > 0$ in the spectral norm. When we are only interested

in a single frequency κ, this is straightforward; we could simply compute an $m \times k$ matrix, where $k < \min(m, n)$, $\mathbf{U}_\kappa$ such that

$$\|\mathbf{G}_\kappa - \mathbf{U}_\kappa \mathbf{U}_\kappa^\dagger \mathbf{G}_\kappa\| < \varepsilon, \tag{4.31}$$

and then use $\mathbf{U}_\kappa \mathbf{U}_\kappa^\dagger \mathbf{G}_\kappa$ as a rank-k approximation. In applications where multiple frequencies must be considered, the straightforward approach is to then compute a new $\mathbf{U}_\kappa$ for each $\mathbf{G}_\kappa$ such that (4.31) holds for each κ of interest. Since this requires computing a new low-rank factorization from scratch for each wavenumber, this can be rather expensive when we are interested in many different wavenumbers.

We will take a different approach in the regime where we are interested in multiple wavenumbers all in an interval $[\kappa_{\min}, \kappa_{\max}]$. Rather than computing a new basis for the range of each $\mathbf{G}_\kappa$, we instead build a single universal basis for the ranges of all $\mathbf{G}_\kappa$ for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$. This amounts to finding a single matrix $\mathbf{U}$ of size $m \times \ell$ such that

$$\|\mathbf{G}_\kappa - \mathbf{U}\mathbf{U}^\dagger \mathbf{G}_\kappa\| < \varepsilon \tag{4.32}$$

for all $\kappa \in [\kappa_{\min}, \kappa_{\max}]$. We can then obtain a rank-ℓ approximation to any $\mathbf{G}_\kappa$ simply by projecting onto the columns of $\mathbf{U}$. For this to be nontrivial, we must place a restriction on the size of ℓ. Remarkably, it turns out that in many cases $\mathbf{U}$ may be chosen such that ℓ is only slightly larger than the maximum ε-rank among all $\mathbf{G}_\kappa$ for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$, which is clearly a lower bound.

We will describe this in more detail in the next section, but for now we consider a concrete example for this geometry. Suppose that we are interested in computing low-rank factorizations of $\mathbf{G}_\kappa$ for $\kappa \in [1, 100]$, so that $\kappa_{\min} = 1$ and $\kappa_{\max} = 100$. We first compute $\mathbf{U} \in \mathbb{C}^{m \times m}$ (how we compute $\mathbf{U}$ is deferred to the next section) and then project the coefficient matrices $\mathbf{G}_\kappa$ onto the leading columns of $\mathbf{U}$. For this to be efficient, we need the leading columns of $\mathbf{U}$ to span roughly the same space as the leading left singular vectors of $\mathbf{G}_\kappa$ for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$. Fig. 4.3(a) plots this projection error for the cases when $\kappa = 1, 50, 100$. Let us temporarily focus on the case when $\kappa = 100$, which is given by the lavender points in Fig. 4.3(a). We see that the error is constant until we project onto approximately $\kappa_{\max}|\Gamma_t|/\pi \approx 43.80$ columns of $\mathbf{U}$. Due to the optimality of the singular value decomposition (SVD) and Fig. 4.2(b), we know the error cannot possibly decay before this point. There is a knee in the curve where the decay begins, and the decay is exponential, just like in the singular values of $\mathbf{G}_{100}$. This tells us that the leading columns of $\mathbf{U}$ span roughly the same space as the leading left singular vectors of $\mathbf{G}_{100}$. This in isolation is not very interesting, but turning our attention to the other points in Fig. 4.3(a), we see that the projection errors for $\mathbf{G}_1$

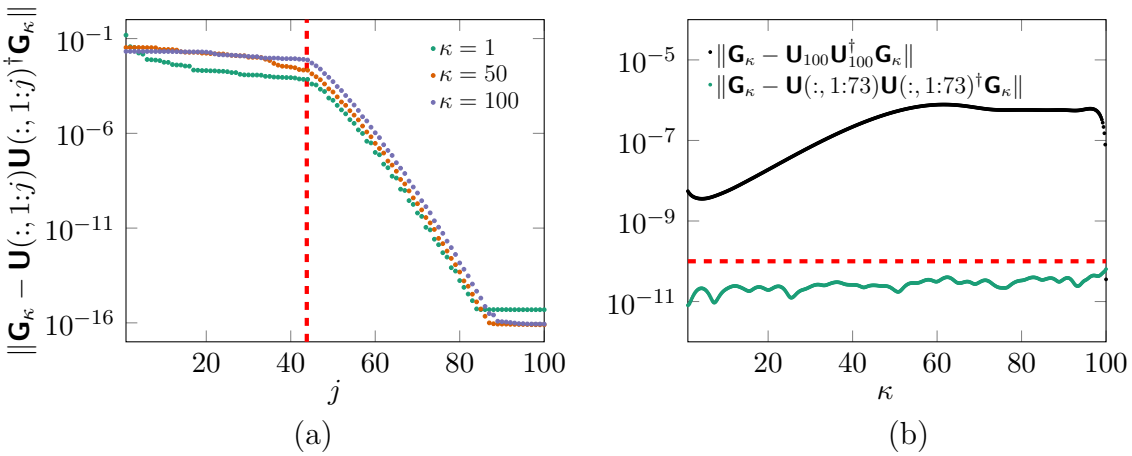


Figure 4.3: (a) Errors in the spectral norm when $\mathbf{G}_1$, $\mathbf{G}_{50}$, and $\mathbf{G}_{100}$ are projected onto the leading columns of $\mathbf{U}$. The red, dashed line demarcates the values $100|\Gamma_t|/\pi \approx 43.80$. (b) Errors in the spectral norm when $\mathbf{G}_\kappa$ is projected onto $\mathbf{U}_{100}$ and $\mathbf{U}(:, 1{:}73)$ for $\kappa \in [1, 100]$. The red, dashed line demarcates the value 10^{-10}.

and $\mathbf{G}_{50}$ behave very similarly, as well. This means that the leading columns of $\mathbf{U}$ also contain information on the leading left singular vectors of $\mathbf{G}_1$ and $\mathbf{G}_{50}$. The error curves for $\mathbf{G}_\kappa$ for other values of $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ would all exhibit the same behavior.

Now suppose $\varepsilon = 10^{-10}$. From Fig. 4.2(b) we see that the corresponding ε-rank of $\mathbf{G}_{100}$ is 70. Let $\mathbf{U}_{100}$ be the matrix whose columns are given by the 70 leading left singular vectors of $\mathbf{G}_{100}$. Fig. 4.3(b) plots the errors in black when $\mathbf{G}_\kappa$ is projected onto $\mathbf{U}_{100}$ for $\kappa \in [1, 100]$. As expected,

$$\|\mathbf{G}_{100} - \mathbf{U}_{100}\mathbf{U}_{100}^\dagger\mathbf{G}_{100}\| < 10^{-10}. \tag{4.33}$$

However, as soon as we leave a small neighborhood of $\kappa = 100$, the projection error increases by a few orders of magnitude. Also shown in Fig. 4.3(b) are the analogous errors when we consider projection onto the first 73 columns of $\mathbf{U}$. Here the errors are below 10^{-10} for all $\kappa \in [1, 100]$. Thus, with only a slightly larger basis, we are able to approximately span not only the range of $\mathbf{G}_{100}$ but also the ranges of $\mathbf{G}_\kappa$ for all $\kappa \in [1, 100]$.

Using a universal basis becomes especially efficient when the ID is used. It turns out that in this case, we do not need to evaluate $\mathbf{U}^\dagger\mathbf{G}_\kappa$ to form a low-rank approximation to $\mathbf{G}_\kappa$. Suppose that $\mathbf{U}$ is an interpolation matrix and $\mathcal{I}$ is the index vector containing

the corresponding skeleton indices in the ID. Then we have the relation

$$\begin{aligned}\|\mathbf{U}\mathbf{G}_\kappa(\mathcal{I},:) - \mathbf{G}_\kappa\| &\le \|\mathbf{U}\mathbf{G}_\kappa(\mathcal{I},:) - \mathbf{U}\mathbf{U}^\dagger\mathbf{G}_\kappa\| + \|\mathbf{U}\mathbf{U}^\dagger\mathbf{G}_\kappa - \mathbf{G}_\kappa\| && (4.34)\\ &\le \|\mathbf{U}\|\|\mathbf{G}_\kappa(\mathcal{I},:) - \mathbf{U}^\dagger\mathbf{G}_\kappa\| + \|\mathbf{U}\mathbf{U}^\dagger\mathbf{G}_\kappa - \mathbf{G}_\kappa\| && (4.35)\\ &= \|\mathbf{U}\|\|\mathbf{G}_\kappa(\mathcal{I},:) - \mathbf{U}(\mathcal{I},:)\mathbf{U}^\dagger\mathbf{G}_\kappa\| + \|\mathbf{U}\mathbf{U}^\dagger\mathbf{G}_\kappa - \mathbf{G}_\kappa\| && (4.36)\\ &\le (1 + \|\mathbf{U}\|)\|\mathbf{U}\mathbf{U}^\dagger\mathbf{G}_\kappa - \mathbf{G}_\kappa\|, && (4.37)\end{aligned}$$

where in (4.36) we have used the fact that $\mathbf{U}(\mathcal{I},:)$ is the $|\mathcal{I}| \times |\mathcal{I}|$ identity. As a result, assuming $\mathbf{U}$ is well-conditioned, we can use $\mathbf{U}\mathbf{G}_\kappa(\mathcal{I},:)$ as the low-rank approximation in place of $\mathbf{U}\mathbf{U}^\dagger\mathbf{G}_\kappa$ without a significant loss in accuracy. The former is clearly substantially cheaper to compute. In this setting we naturally refer to our universal basis as *universal skeletons*.

4.3.3 Universal skeletons

In this section we will more formally expand on the notion of universal bases introduced in the previous section. We first return to the prior example in slightly more detail. For every integer $N > 1$, we define the matrix

$$\mathbf{G}^{(N)}_{[\kappa_{\min},\kappa_{\max}]} = \begin{bmatrix}\mathbf{G}_{\kappa_1^{(N)}} & \mathbf{G}_{\kappa_2^{(N)}} & \dots & \mathbf{G}_{\kappa_N^{(N)}}\end{bmatrix} \in \mathbb{C}^{m\times Nn} \tag{4.38}$$

where $\{\kappa_j^{(N)}\}_{j=1}^N$ are the N Chebyshev points of the second kind in $[\kappa_{\min}, \kappa_{\max}]$ ordered increasingly. It is also convenient to define $\mathbf{G}^{(1)}_{[\kappa_{\min},\kappa_{\max}]} = \mathbf{G}_{\kappa_{\max}}$. For fixed N and $\varepsilon > 0$, suppose $\mathbf{U}_\varepsilon^{(N)} \in \mathbb{C}^{m\times \ell_\varepsilon^{(N)}}$ satisfies

$$\|\mathbf{G}^{(N)}_{[\kappa_{\min},\kappa_{\max}]} - \mathbf{U}_\varepsilon^{(N)}(\mathbf{U}_\varepsilon^{(N)})^\dagger\mathbf{G}^{(N)}_{[\kappa_{\min},\kappa_{\max}]}\| < \varepsilon \tag{4.39}$$

where $\ell_\epsilon^{(N)} < \min(m,n)$ is the rank of the approximation. Then it is necessarily true that

$$\|\mathbf{G}_{\kappa_j^{(N)}} - \mathbf{U}_\varepsilon^{(N)}(\mathbf{U}_\varepsilon^{(N)})^\dagger\mathbf{G}_{\kappa_j^{(N)}}\| < \varepsilon, \quad j = 1,2,\dots,N. \tag{4.40}$$

We could hope that for sufficiently large N, (4.39) further implies that

$$\|\mathbf{G}_\kappa - \mathbf{U}_\varepsilon^{(N)}(\mathbf{U}_\varepsilon^{(N)})^\dagger\mathbf{G}_\kappa\| < C\varepsilon \tag{4.41}$$

where C is a constant close to 1 for all $\kappa \in [\kappa_{\min}, \kappa_{\max}]$.

We will temporarily restrict ourselves again to the case where $\kappa_{\min} = 1$, $\kappa_{\max} = 100$, and $\varepsilon = 10^{-10}$. We compute $\mathbf{U}_\varepsilon^{(N)} \in \mathbb{C}^{m\times\ell_\varepsilon^{(N)}}$ by truncating the SVD of $\mathbf{G}^{(N)}_{[1,100]}$ for $N = 1,3,5,7$, and then examine the projection errors, given by the left-hand side of (4.41), as a function of κ for each choice of N. Fig. 4.4(b) plots the resulting errors for

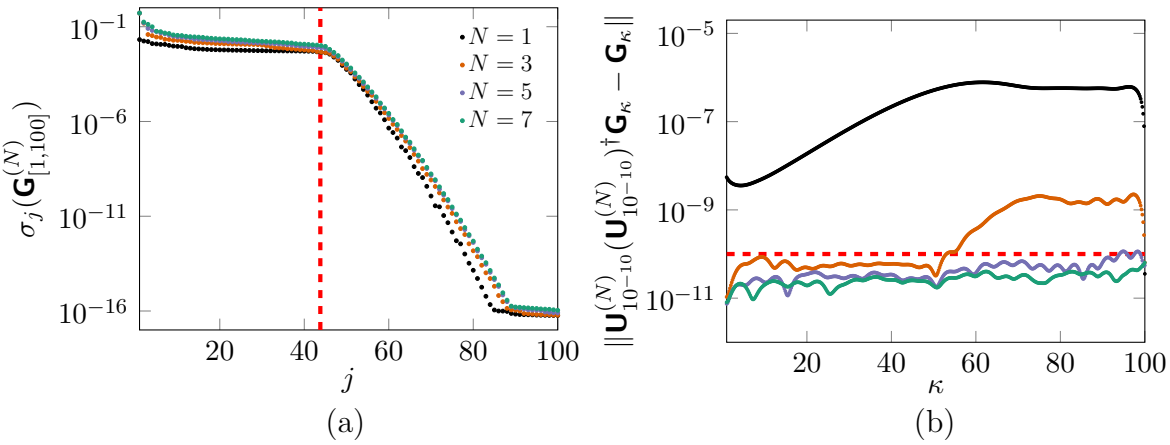


Figure 4.4: (a) First 100 singular values of $\mathbf{G}^{(N)}_{[1,100]}$ for $N = 1, 3, 5, 7$ computed in double precision. The red, dashed line demarcates the value $100|\Gamma_t|/\pi \approx 43.80$. (b) Errors in the spectral norm when $\mathbf{G}_\kappa$ is projected onto the columns of $\mathbf{U}^{(N)}_\varepsilon$ with $\varepsilon = 10^{-10}$ and $N = 1, 3, 5, 7$. The red, dashed line demarcates the value 10^{-10}.

500 different values of κ in the interval $[1, 100]$. As we saw in the preceding section, the columns of $\mathbf{U}^{(1)}_{10^{-10}}$ form a good basis for the range of $\mathbf{G}_{100}$ up to the desired accuracy of 10^{-10}, but this is not true in general for the ranges of $\mathbf{G}_\kappa$ for $\kappa < 100$. In fact, an error less than 10^{-10} is achieved at no other wavenumber outside of a small neighborhood of $\kappa = 100$ with up to four digits lost for some choices of κ. The situation is substantially improved when the columns of $\mathbf{U}^{(3)}_{10^{-10}}$ are taken as a universal basis. An error of less than 10^{-10} is again obtained near $\kappa = 100$ but now also for all values of $\kappa \in [1, 52]$. Moreover, no more than two digits are lost across the entire interval. The situation improves further with $\mathbf{U}^{(5)}_{10^{-10}}$. Now the errors are below 10^{-10} for most $\kappa \in [1, 100]$, and we lose at most one digit. Finally, with $\mathbf{U}^{(7)}_{10^{-10}}$ the errors are below 10^{-10} for all $\kappa \in [1, 100]$. Indeed, we took $\mathbf{U} = \mathbf{U}^{(7)}_{10^{-10}}$ in the previous section. While we have chosen a specific value of $\varepsilon = 10^{-10}$ here, the behavior for other values of ε down to near machine precision would be similar with $N = 7$ sufficing to control the errors in the entire interval $[1, 100]$.

For the low-rank approximation in (4.41) to be useful, we must put some restriction on the size of $\ell^{(N)}_\varepsilon$. Clearly, $\ell^{(N)}_\varepsilon$ must be bounded below by the ε-rank of $\mathbf{G}^{(N)}_{[\kappa_{\min},\kappa_{\max}]}$, which is defined by

$$k^{(N)}_\varepsilon = \max\{j \in \mathbb{N} : \sigma_j(\mathbf{G}^{(N)}_{[\kappa_{\min},\kappa_{\max}]}) \geq \varepsilon\}. \tag{4.42}$$

Furthermore, taking the columns of $\mathbf{U}^{(N)}_\varepsilon$ to span the same space as the leading $k^{(N)}_\varepsilon$ left singular vectors of $\mathbf{G}^{(N)}_{[\kappa_{\min},\kappa_{\max}]}$ guarantees that (4.39) holds with $\ell^{(N)}_\varepsilon = k^{(N)}_\varepsilon$, and

so it suffices to study $k_\varepsilon^{(N)}$. A necessary condition for (4.41) to hold is

$$k_\varepsilon^{(N)} \geq \sup_{\kappa \in [\kappa_{\min}, \kappa_{\max}]} \min \{j \in \mathbb{N} : \sigma_j(\mathbf{G}_\kappa) \geq \varepsilon\}, \tag{4.43}$$

and from the earlier estimates, we typically expect the supremum in the right-hand side of (4.43) to be attained when $\kappa = \kappa_{\max}$. As a result, the most we can usually hope for is

$$k_\varepsilon^{(N)} \geq \widehat{C} \min \{j \in \mathbb{N} : \sigma_j(\mathbf{G}_{\kappa_{\max}}) \geq \varepsilon\} \tag{4.44}$$

for some constant $\widehat{C}$ only slightly larger than 1.

Fig. 4.4(a) plots the singular values of $\mathbf{G}_{[1,100]}^{(N)}$ for $N = 1, 3, 5, 7$. It is interesting to note that the singular values of $\mathbf{G}_{[1,100]}^{(N)}$ for $N = 3, 5, 7$ all behave similarly to the singular values of $\mathbf{G}_{100}$. In particular, we see that there is a knee after which the singular values of all the matrices exponentially decay, and moreover this knee occurs close to that of $\mathbf{G}_{100}$. The rate of exponential decay of the singular values for each matrix is also close to that of $\mathbf{G}_{100}$. Therefore, it is unsurprising that the ε-ranks of $\mathbf{G}_{[1,100]}^{(N)}$ for $N > 1$ are not actually much higher than that of $\mathbf{G}_{100}$. In fact, we see that $k_{10^{-10}}^{(1)} = 70$, $k_{10^{-10}}^{(3)} = 72$, $k_{10^{-10}}^{(5)} = 73$, and $k_{10^{-10}}^{(7)} = 73$. It is especially interesting to note that $k_{10^{-10}}^{(5)} = k_{10^{-10}}^{(7)}$. This suggests that while projection onto $\mathbf{U}_{10^{-10}}^{(7)}$ yields higher low-rank approximation errors at the values $\{\kappa_j^{(5)}\}_{j=1}^5$ than projection onto $\mathbf{U}_{10^{-10}}^{(5)}$, it yields an overall lower worst-case error across the entire interval.

So far, we have considered a specific geometry and interval of wavenumbers, but the observed behavior is illustrative of a more general phenomenon. Indeed, extensive numerical evidence suggests that there is a point past which the singular values of $\mathbf{G}_{[\kappa_{\min}, \kappa_{\max}]}^{(N)}$ will exponentially decay, and it will be approximately upper bounded by

$$\frac{\kappa_{\max} |\Gamma_t|}{\pi} \tag{4.45}$$

for all N. More evidence of this is shown in Fig. 4.5. The left images in Fig. 4.5 show four more configurations of source and target curves, shown in red and blue, respectively. For each geometry we consider the interval of wavenumbers $[1, \kappa_{\max}]$, where $\kappa_{\max}$ is shown in the top right corner of each plot, along with the value of $|\Gamma_t|$. The images in the right column show the singular values of the corresponding $\mathbf{G}_{\kappa_{\max}}$ and $\mathbf{G}_{[\kappa_{\min}, \kappa_{\max}]}^{(N)}$ matrices where N was chosen such that $\mathbf{U}_\varepsilon^{(N)}$ computed using the SVD satisfies (4.41) for all $\kappa \in [1, \kappa_{\max}]$ with $\varepsilon = 10^{-10}$. In all cases the singular values of $\mathbf{G}_{\kappa_{\max}}^{(N)}$ start to exponentially decay around $\kappa_{\max}|\Gamma_t|/\pi$.

We are unfortunately not able to prove a formal result to this effect, so we will summarize our observations rather imprecisely as follows. Let $\Gamma_s \subset \mathbb{R}^2$ and $\Gamma_t \subset \mathbb{R}^2$

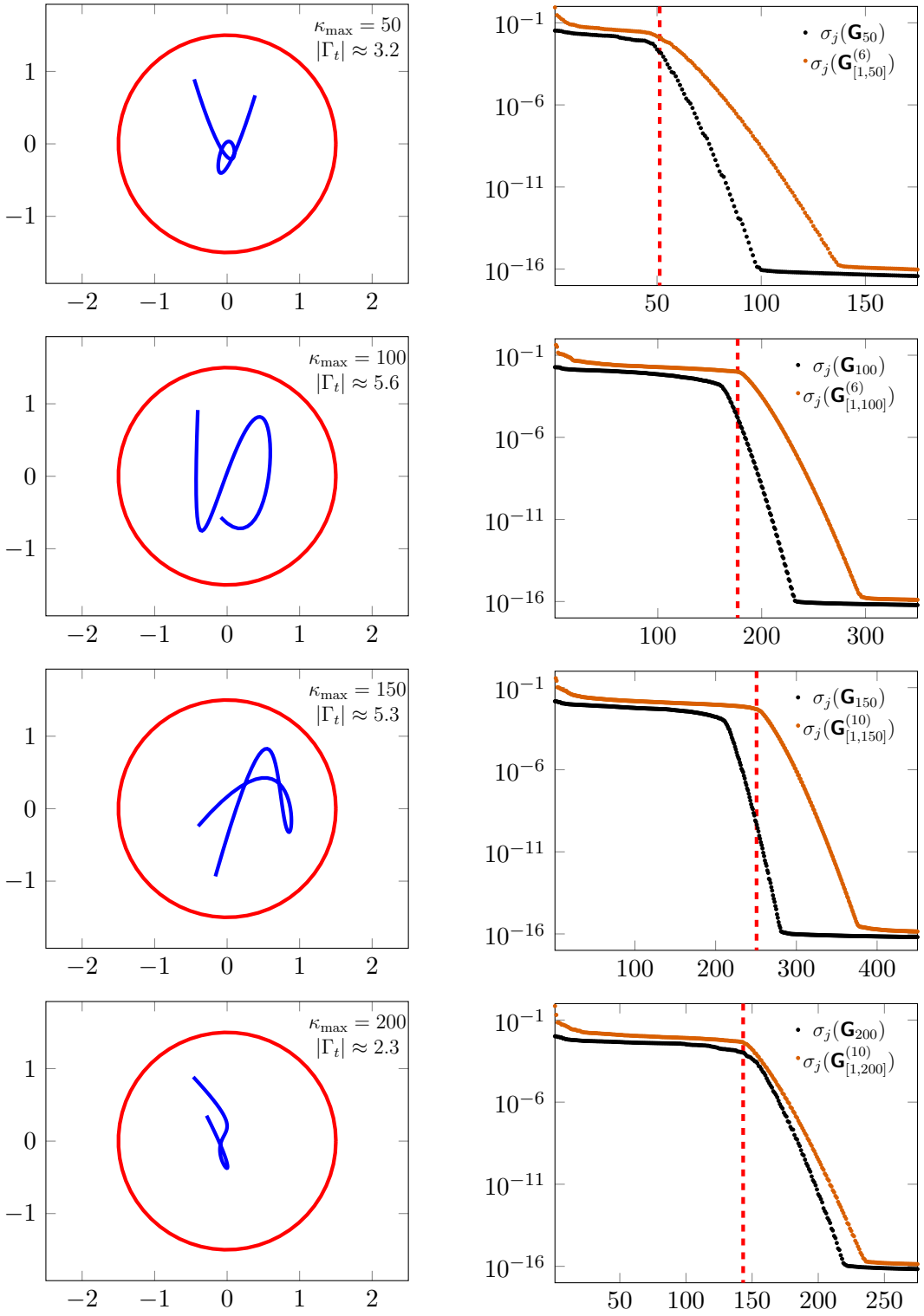


Figure 4.5: The left column shows the four geometries with Γ_s denoted in red and Γ_t denoted in blue. The right column shows the singular values of $\mathbf{G}_{\kappa_{\max}}$ and $\mathbf{G}^{(N)}_{[\kappa_{\min},\kappa_{\max}]}$ for the smallest N sufficient for (4.41) to hold with $\varepsilon = 10^{-10}$ for all $\kappa \in [\kappa_{\min}, \kappa_{\max}]$. The red, dashed line demarcates the value $\kappa_{\max}|\Gamma_t|/\pi$.

be Lipschitz curves that are sufficiently separated, and $\kappa_{\min}, \kappa_{\max} \in \mathbb{R}$ be given such that $\kappa_{\max} > \kappa_{\min} > 0$. Suppose that for all $\kappa \in [\kappa_{\min}, \kappa_{\max}]$, $\mathbf{G}_\kappa \in \mathbb{C}^{m \times n}$ is an appropriate collocation discretization of the operator (4.26) in the manner described in Section 4.2.1. Then we can find a matrix $\mathbf{U} \in \mathbb{C}^{m \times m}$ such that the projection error

$$\sup_{\kappa \in [\kappa_{\min}, \kappa_{\max}]} \|\mathbf{G}_\kappa - \mathbf{U}(:, 1:\ell)\mathbf{U}(:, 1:\ell)^\dagger \mathbf{G}_\kappa\| \tag{4.46}$$

starts to decay exponentially as a function of ℓ when

$$\ell \approx \frac{\kappa_{\max}|\Gamma_t|}{\pi}. \tag{4.47}$$

We have been intentionally vague about what sufficient separation between source and target curves entails, as we are not aware of a precise result on this even in the single wavenumber case. Empirically, we find that it typically suffices for the curves to be separated by about one wavelength (when the wavenumber is $\kappa_{\max}$), except in small neighborhoods of a finite number of isolated points. For example, Γ_s and Γ_t may be pieces of the same contour in 2D with touching endpoints. As seen in our numerical experiments, the relation in (4.47) is not an exact equality, and so this should be considered more of a heuristic in the same vein as those introduced for the single wavenumber case in Section 4.3.1. It is worth noting that (4.47) agrees with the second heuristic discussed in Section 4.3.1, which was based on a bandlimit argument, when $\kappa = \kappa_{\max}$ assuming $|\Gamma_s| \geq |\Gamma_t|$. The ramification of this is that in a broad range of cases, the value of ℓ required to get (4.47) to be less than ε will not be much greater than the ε-rank of $\mathbf{G}_{\kappa_{\max}}$. Consequently, in these cases we can achieve (4.44) with $\widehat{C}$ close to 1. Of course when $|\Gamma_s| < |\Gamma_t|$, we could build a universal basis for the row space of $\mathbf{G}_\kappa$ instead.

In certain special cases the right-hand side of (4.47) is suboptimal and overestimates the point at which the exponential decay will begin. One such instance is illustrated in Fig. 4.6(a), which considers the case where the target and source curves correspond to the left and right halves, respectively, of an ellipse with semi-major axis length 5 and semi-minor axis length 1. Fig. 4.6(b) shows the corresponding singular value plot when $\kappa_{\max} = 50$. The knee in the singular values of $\mathbf{G}^{(27)}_{[1,50]}$ is now at about $\kappa_{\max}|\Gamma_t|/(2\pi) \approx 83.60$ (denoted by the dashed red line in Fig. 4.6(b)), which is half of what is predicted by the right-hand side of (4.47). This is due to the fact that the curves are configured such that only scattered fields with a certain directionality can be produced on Γ_t by sources on Γ_s. We will explore this in more detail in the next section. This example also illustrates that sometimes it is not possible to achieve (4.44)

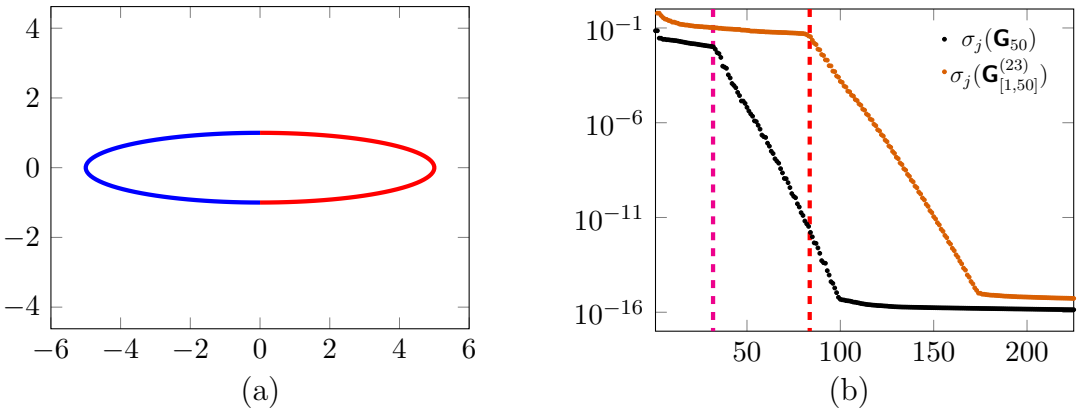


Figure 4.6: (a) Geometry of a problem where Γ_s (denoted in red) and Γ_t (denoted in blue) correspond to halves of an ellipse. (b) Corresponding singular values computed for this problem when $\kappa_{\max} = 50$. The red and magenta, dashed lines correspond to the values $|\Gamma_t|\kappa_{\max}/(2\pi) \approx 83.60$ and $2\kappa_{\max}/\pi \approx 31.83$, respectively.

with $\widehat{C}$ close to 1. Due to the elongated nature of this geometry, the behavior of the singular values of $\mathbf{G}_{\kappa_{\max}}$ is more tightly controlled by the first heuristic in Section 4.3.1, which was based on the angle of interaction between the sources and targets. As such, the knee in the singular values of $\mathbf{G}_{\kappa}$ occurs around $2\kappa/\pi$ (denoted by the magenta, dashed line in Fig. 4.6(b)), which corresponds to two times the height of the ellipse measured in wavelengths [114].

We could have also formulated our observations in terms of the continuum operator (4.26), at the expense of a slightly more cumbersome notation. In this case the existence of a universal basis for a range of wavenumbers $[\kappa_{\min}, \kappa_{\max}]$ amounts to the statement that there exists a collection of functions $\{u^{(j)}\}_{j=1}^{\infty} \subset L^2(\Gamma_t)$ such that for any $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ there exists $\{v_\kappa^{(j)}\}_{j=1}^{\infty} \subset L^2(\Gamma_s)$ for which the error

$$\left\| \mathcal{S}_\kappa - \sum_{j=1}^{\ell} u^{(j)} (v_\kappa^{(j)})^* \right\| \tag{4.48}$$

decays exponentially after $\ell \approx \kappa|\Gamma_t|/\pi$.

Remark 4.1. For a few special geometries, a low-rank factorization of the continuum operator (4.26) can be worked out explicitly. For example, suppose R_s and R_t are positive real numbers with $R_s > 2R_t$, and let $\Gamma_s = \{\boldsymbol{x} \in \mathbb{R}^2 : \|\boldsymbol{x}\| = R_s\}$ and $\Gamma_t = \{\boldsymbol{x} \in \mathbb{R}^2 : \|\boldsymbol{x}\| = R_t\}$ be concentric circles. In this case it follows from Graf's addition theorem [1, (9.1.79)] that for any $\kappa > 0$, $\mathcal{S}_\kappa$ in (4.26) is equivalent to

$$[\mathcal{S}_\kappa \sigma](\theta) = \frac{iR_s}{4} \sum_{j=-\infty}^{\infty} H_j(\kappa R_s) J_j(\kappa R_t) e^{ij\theta} \int_0^{2\pi} e^{-ij\varphi} \sigma(R_s e^{ij\varphi})\, d\varphi \tag{4.49}$$

where $\boldsymbol{x} = (R_t \cos\theta, R_t \sin\theta) \in \Gamma_t$ and any $\sigma \in L^2(\Gamma_s)$. From (4.49) it follows that we can take

$$u^{(j)}(\boldsymbol{x}) = e^{ij\theta}, \quad \boldsymbol{x} = (R_t \cos\theta, R_t \sin\theta) \tag{4.50}$$

and

$$v_\kappa^{(j)}(\boldsymbol{y}) = \frac{iR_s}{4} H_j(\kappa R_s) J_j(\kappa R_t) e^{ij\varphi}, \quad \boldsymbol{y} = (R_s \cos\varphi, R_s \sin\varphi) \tag{4.51}$$

for $j = 1, 2, \ldots$. From standard asymptotics for Bessel and Hankel functions, we expect the approximation error in (4.49) to be approximately ε when the summation is truncated at $\pm\ell$ when $\ell \approx \kappa_{\max} R_t + \mathcal{O}(\log(1/\varepsilon)^{2/3})$ [39, 141].

Remark 4.2. This discussion extends naturally to three dimensions. Here $\Gamma_s \subset \mathbb{R}^3$ and $\Gamma_t \subset \mathbb{R}^3$ are surfaces, and the single layer operator is given by

$$[\mathcal{S}_\kappa \sigma](\boldsymbol{x}) = \frac{1}{4\pi} \int_{\Gamma_s} \frac{e^{i\kappa\|\boldsymbol{x}-\boldsymbol{y}\|}}{\|\boldsymbol{x}-\boldsymbol{y}\|} \sigma(\boldsymbol{y})\, ds(\boldsymbol{y}), \quad \boldsymbol{x} \in \Gamma_t \tag{4.52}$$

for $\sigma \in L^2(\Gamma_s)$. Fig. 4.7 considers the case when

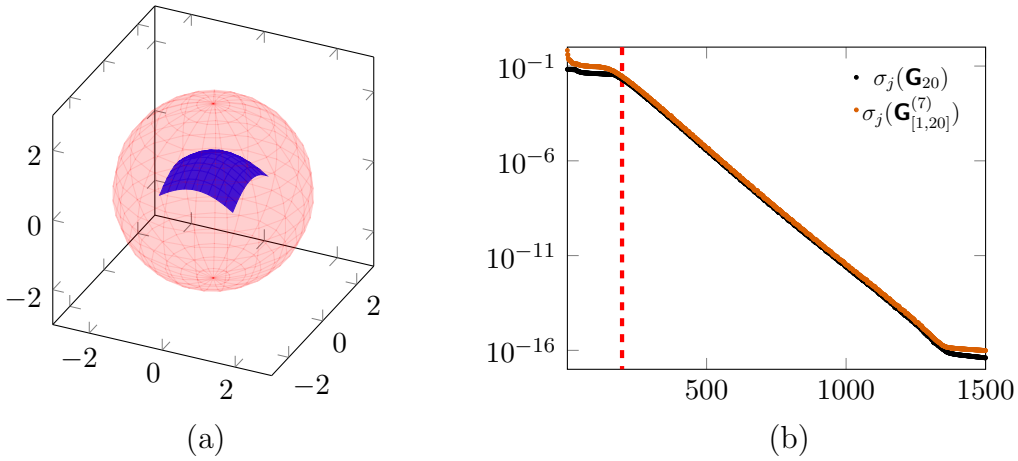
$$\Gamma_t = \{(u, v, \cos(\sqrt{u^2+v^2}) : u, v \in [-1, 1]\} \tag{4.53}$$

and Γ_s is the sphere of radius 2.5 centered at the origin. The geometry is plotted in Fig. 4.7(a). The singular values of $\mathbf{G}_{20}$ and $\mathbf{G}_{[1,20]}^{(7)}$ ($\mathbf{U}_\varepsilon^{(7)}$ obtained by truncating the SVD of $\mathbf{G}_{[1,20]}^{(7)}$ satisfies (4.41) with $\varepsilon = 10^{-10}$ for all $\kappa \in [1, 20]$), which now correspond to discretizations of (4.52), are plotted in Fig. 4.7(b). The red, dashed line indicates the value $\kappa_{\max}^2 |\Gamma_t|/\pi^2 \approx 196.98$, which is now seen to be a good approximation to the knees in the singular value curves of both $\mathbf{G}_{20}$ and $\mathbf{G}_{[1,20]}^{(7)}$.

Again, the latter fact seems to hold more generally. We summarize our numerical experience as follows. Let $\Gamma_s \subset \mathbb{R}^3$ and $\Gamma_t \subset \mathbb{R}^3$ be smooth and separated surfaces, and $\kappa_{\min}, \kappa_{\max} \in \mathbb{R}$ be given such that $\kappa_{\max} > \kappa_{\min} > 0$. Further assume that Γ_t is path connected. Suppose that for all $\kappa \in [\kappa_{\min}, \kappa_{\max}]$, $\mathbf{G}_\kappa \in \mathbb{C}^{m \times n}$ is an appropriate collocation discretization of the operator (4.52) in the manner described in Section 4.2.1. Then there exists $\mathbf{U} \in \mathbb{C}^{m \times m}$ such that (4.46) starts to decay exponentially as a function of ℓ when

$$\ell \approx \max\left(\frac{\kappa_{\max}^2 |\Gamma_t|}{\pi^2}, \frac{\kappa_{\max} \delta(\Gamma_t)}{\pi}\right) \tag{4.54}$$

where $\delta(\Gamma_t)$ is the supremum over the lengths of the minimal geodesics on Γ_t. The second term in the maximum in the right-hand side of (4.54) accounts for the case when Γ_t is "thin", and therefore the behavior is more like that of a one-dimensional curve than a surface. It is worth pointing out that unlike in 2D where the right-hand side



(a) (b)

Figure 4.7: (a) Geometry of a 3D problem where Γ_t is a surface (denoted in blue) and Γ_s is a sphere (denoted in red). (b) Singular values of $\mathbf{G}_{20}$ and $\mathbf{G}^{(7)}_{[1,20]}$ corresponding to this geometry. The red, dashed line is located at $\kappa^2_{\max}|\Gamma_t|/\pi^2 \approx 196.98$.

(4.47) is only significantly suboptimal in certain special cases, (4.54) quite frequently overestimates the necessary size of a universal basis for reasons that will be discussed in the next section.

4.3.4 Bandlimited fields

To motivate our observations on universal bases, we first consider the one-dimensional Helmholtz equation given by

$$-u''(x) - \kappa^2 u(x) = 0, \quad x \in (-1, 1) \tag{4.55}$$

for some $\kappa > 0$. Obviously, any smooth solution of (4.55) is given by

$$u(x) = \alpha_1 e^{i\kappa x} + \alpha_2 e^{-i\kappa x} \tag{4.56}$$

for some $\alpha_1, \alpha_2 \in \mathbb{C}$. For fixed $c > 0$ and any $\kappa \leq c$, clearly all functions of the form (4.56) may be written

$$u(x) = \int_{-1}^{1} e^{icx\xi} \, d\mu(\xi) \tag{4.57}$$

where μ is a sum of Dirac measures supported on $\pm\kappa/c$, and the total variation of μ is controlled by the magnitude of the coefficients α_1 and α_2. We recall that (4.57) is precisely the definition of a bandlimited function stated in Section 4.2.3. Therefore, the solutions to (4.55) for all $\kappa \in (0, c]$ are bandlimited with bandlimit c. As discussed in Section 4.2.3, the PSWFs are a natural basis for approximating such functions. This motivates using the PSWFs as a universal basis for the ranges of families of

solution operators to (4.55) for any $\kappa \in (0, c]$, such as the maps that take Dirichlet values at -1 and 1 to the corresponding solution. Combining Conjecture 4.2 and Lemma 4.1 then shows us that the error in this approximation decays exponentially with the number of PSWFs after there are approximately $2c/\pi$ terms.

In higher dimensions the relation between scattered fields and bandlimit is more subtle and has been studied by a number of authors (e.g., [31, 32, 39, 149, 165]). We consider a situation in two or three dimensions where the target domain Γ_t lies in the unit ball $D = \{\boldsymbol{x} : \|\boldsymbol{x}\| \leq 1\}$, which is disjoint from the source domain Γ_s (i.e., $\Gamma_t \subset D$ and $\Gamma_s \subset D^c$). In this case for any $\kappa > 0$ and density function $q \in L^2(\Gamma_s)$, we can obtain a field given by

$$u(\boldsymbol{x}) = [\mathcal{S}_\kappa q](\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma_t \tag{4.58}$$

where $\mathcal{S}_\kappa$ is given by (4.26) or (4.52). The field u can be extended to a solution of the homogeneous Helmholtz equation on all of D. Since linear combinations of plane waves are dense in this space (see, e.g., [117, 121, 146]), we can approximate u to an arbitrary accuracy by a function of the form

$$\widetilde{u}(\boldsymbol{x}) = \sum_{j=1}^{M} \alpha_j e^{i\kappa \boldsymbol{d}_j \cdot \boldsymbol{x}}, \quad \boldsymbol{x} \in \Gamma_t \tag{4.59}$$

where $\{\boldsymbol{d}_j\}_{j=1}^M$ are unit vectors and $\{\alpha_j\}_{j=1}^M \subset \mathbb{C}$ for some $M \in \mathbb{N}$. For any $c \geq \kappa$, we can express (4.59) as

$$\widetilde{u}(\boldsymbol{x}) = \int_D e^{ic\boldsymbol{\xi} \cdot \boldsymbol{x}} \, d\mu(\boldsymbol{\xi}), \quad \boldsymbol{x} \in \Gamma_t \tag{4.60}$$

where μ is supported on the circle in 2D and the sphere in 3D of radius κ/c. The total variation of μ is controlled by the norm of q and the distance between Γ_s and D. Thus, fields on Γ_t generated by densities on Γ_s may be viewed as multi-dimensional bandlimited functions, with bandlimit c, restricted to Γ_t.

The relevant truncated Fourier operator now goes from $L^2(D)$ to $L^2(\Gamma_t)$ and is given by

$$[\mathcal{F}_c \sigma](\boldsymbol{x}) = \int_D e^{ic\boldsymbol{\xi} \cdot \boldsymbol{x}} \, d\boldsymbol{\xi}, \quad \boldsymbol{x} \in \Gamma_t. \tag{4.61}$$

As mentioned in Section 4.2.3, the generalization of the PSWFs in higher dimensions are the GPSFs. However, the vast majority of literature on GPSFs treat the case where the bandlimited function is restricted to a disk. Notable exceptions are Section 3 in [138], which considers some abstract scenarios, and the paper [21], where they consider restrictions of bandlimited functions to rectangles. In fact, we are not aware of any literature that extends GPSFs to settings where the domain and codomain

spaces are not of the same dimension, as in our case. Obviously, we must use the leading left singular functions of (4.61) in lieu of eigenfunctions for the expansion, and the number of terms necessary to attain a fixed accuracy depends on the corresponding singular values in lieu of eigenvalues.

Establishing a rigorous bound on the behavior of (4.61) is outside the scope of this work. Fortunately, we can provide a simple heuristic, based on the Shannon–Nyquist theorem. Since the bandlimit is c and the field is restricted to a domain of size $|\Gamma_t|$, the standard two points per wavelength rule tells us that in two dimensions, we should expect roughly $c|\Gamma_t|/\pi$ singular vectors to be required to resolve the oscillations, and the singular values corresponding to the remaining modes to decay. We can prove this for specific geometries, such as when Γ_t is a line segment. In this case the above statement follows from the fact that (4.61) simplifies to the one-dimensional truncated Fourier integral operator when, for example, an arc length parametrization is used. An abundance of numerical evidence suggests that this holds for other geometries as well. This is illustrated in Fig. 4.8, where each row considers a case where Γ_t is given by the curve in the left image of the respective row of Fig. 4.5 and $c = \kappa_{\max}$. In the left images of Fig. 4.8, the singular values of a discretization $\mathbf{F}_c$ of $\mathcal{F}_c$ are plotted. The right images of Fig. 4.8 plot the error

$$\text{err}_j = \|\mathbf{G}^{(N)}_{[1,\kappa_{\max}]} - \mathbf{U}(:, 1:j-1)\mathbf{U}(:, 1:j-1)^* \mathbf{G}^{(N)}_{[1,\kappa_{\max}]}\| \tag{4.62}$$

where the columns of $\mathbf{U}$ are the left singular vectors of $\mathbf{F}_c$ and for the values of N and $\kappa_{\max}$ considered in the respective rows in Fig. 4.5. In every case err_j decays exponentially shortly after j exceeds $c|\Gamma_t|/\pi$. In three dimensions applying the Shannon–Nyquist idea in each dimension yields the estimate $c^2|\Gamma_t|/\pi^2$. However, we believe that this estimate is rather loose and for there to be better estimates depending on more properties of Γ_t, analogous to Petersen-Middleton theorems in classical sampling theory [128]. All the ideas here are closely related to sampling theory, specifically the mobile sensing problem [155] and MRI reconstruction [123].

We also note that the ε-rank of (4.65) must be bounded from above by the ε-rank of the truncated Fourier operator from $L^2(D)$ to itself given by (4.23). From (4.25) it is clear that this operator has about $c^2/4$ "non-decaying" singular values in two dimensions. Thus, this is the more relevant heuristic in cases in two dimensions when $|\Gamma_t| > c\pi/4$. This is illustrated in Fig. 4.9, which considers the case where Γ_t is a "swirl" as shown in Fig. 4.9(a). For this curve $|\Gamma_t| \approx 10.06$ so when $c = 10$, the arc length heuristic would suggest that the $\sigma_j(\mathbf{F}_{10})$ start to decay when $j \approx 63.86$ (denoted by the red, dashed line), when in fact a better heuristic is $j \approx c^2/4 = 25$ (denoted

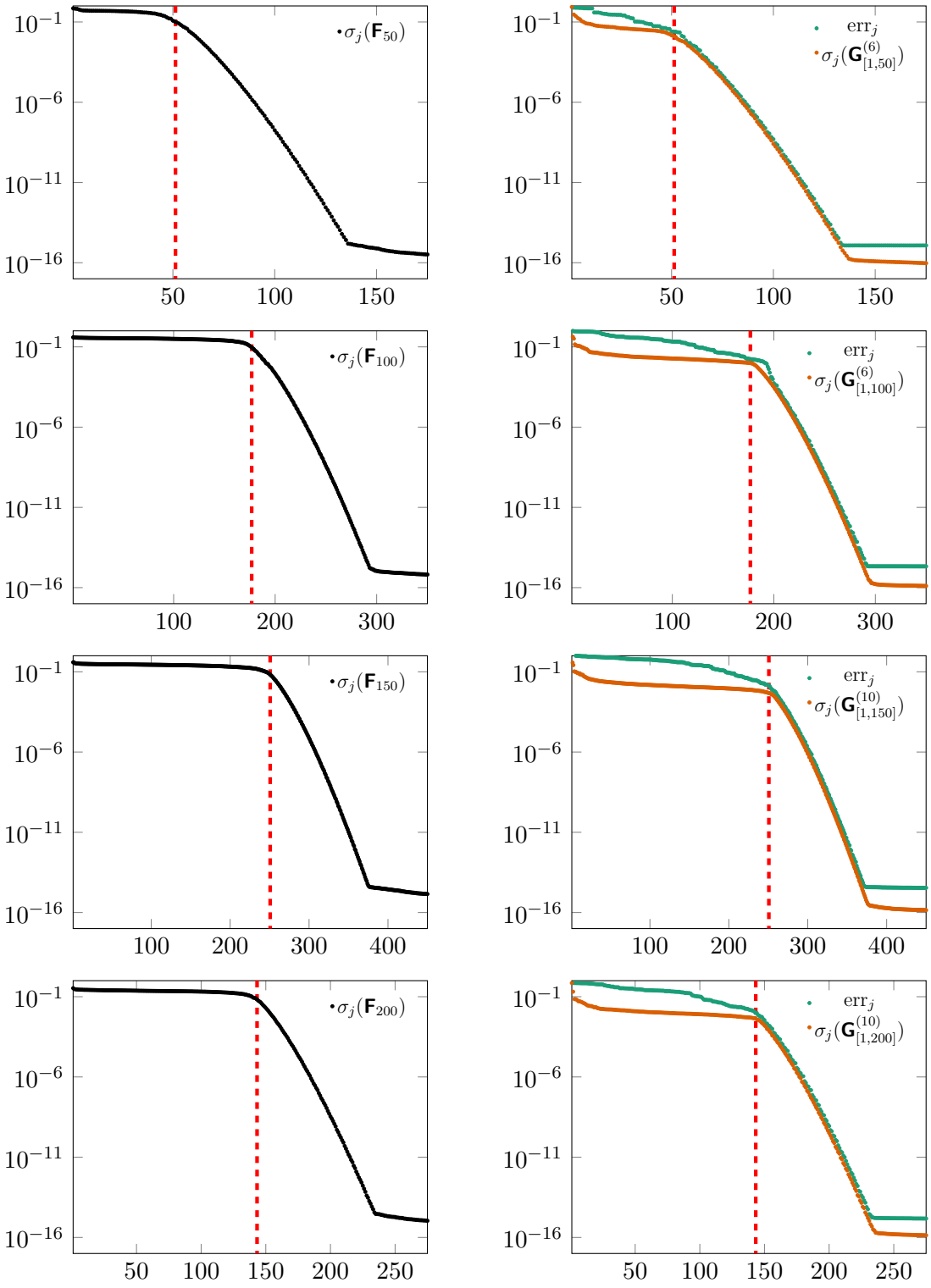


Figure 4.8: The left plots show the singular values of $\mathbf{F}_c$ for the geometries and $c = \kappa_{\max}$ in the respective rows of Fig. 4.5. The right plots show the singular values of $\mathbf{G}^{(N)}_{[1,\kappa_{\max}]}$, again taken from the respective plot in Fig. 4.5, and err_j as defined in (4.62). The red, dashed lines demarcate the value $c|\Gamma_t|/\pi$ for the corresponding parameters in every image.

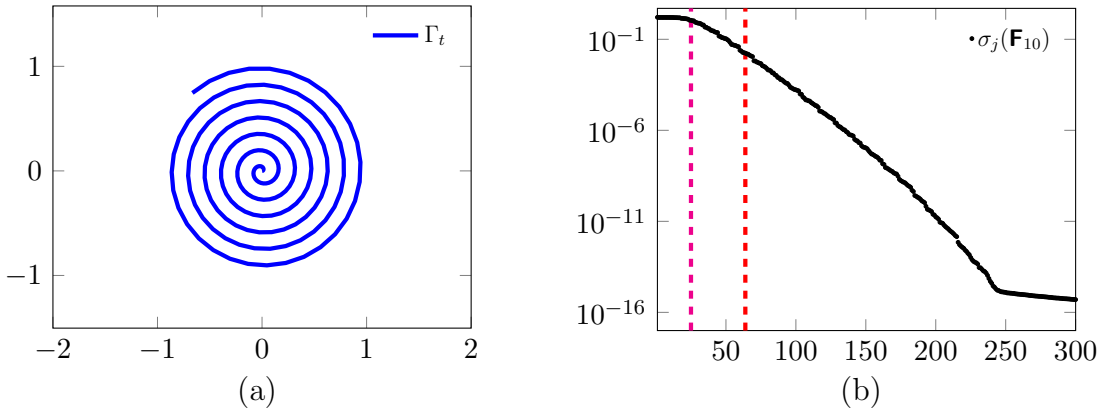


Figure 4.9: (a) Swirl geometry. Note that $|\Gamma_t| = 10.06$, whereas $\operatorname{diam}\Gamma_t < 1$. (b) Singular values of $\mathbf{F}_c$ for this geometry. The red and magenta, dashed lines denote the value $c|\Gamma_t|/\pi \approx 63.86$ and $c^2/4 = 25$.

by the magenta, dashed line). In three dimensions it follows from (4.25) that the analogous estimate is $2c^3/(9\pi)$.

Remark 4.3. Let us consider a generic truncated Fourier operator $\mathcal{F}_c : L^2(S) \to L^2(\Gamma_t)$ given by

$$[\mathcal{F}_c\sigma](\boldsymbol{x}) = \int_S e^{ic\boldsymbol{\xi}\cdot\boldsymbol{x}}\sigma(\boldsymbol{x})\,d\boldsymbol{x}, \quad \boldsymbol{x} \in \Gamma_t \tag{4.63}$$

for some $S \subset \mathbb{R}^2$ and 1D curve Γ_t. By a simple change of variables, we may shift S anywhere in $\mathbb{R}^2$. Thus, the point at which the singular values of $\mathcal{F}_c$ transition to exponentially decaying can be approximated using only the diameter of S. From the preceding discussion, we expect this point to be around $|\Gamma_t| \operatorname{diam} S/(2\pi)$. In cases such as in Fig. 4.6, the interactions between the source and target curves have a strong directionality, so the relevant S is an acute sector with radius $c = \kappa_{\max}$. The diameter of such a sector is $c/2$, which explains why the relevant heuristic for Fig. 4.6 is only half of the leading term in (4.47).

Remark 4.4. Both heuristics mentioned in Section 4.3.1 can actually be derived from (4.60). Suppose we are in the two-dimensional case. Since for a single frequency μ in (4.60) is supported on a circle, we can express $\widetilde{u}$ in (4.59) as

$$\widetilde{u}(\boldsymbol{x}) = \int_0^{2\pi} e^{i\kappa(\cos\theta,\,\sin\theta)\cdot\boldsymbol{x}}\,d\mu(\theta), \quad \boldsymbol{x} \in \Gamma_t \tag{4.64}$$

where now μ is a measure on $[0, 2\pi]$. The relevant truncated Fourier operator $\mathcal{F}_c : L^2([0,2\pi]) \to L^2(\Gamma_t)$ then becomes

$$[\mathcal{F}_c\sigma](\boldsymbol{x}) = \int_0^{2\pi} e^{i\kappa(\cos\theta,\,\sin\theta)\cdot\boldsymbol{x}}\,d\theta, \quad \boldsymbol{x} \in \Gamma_t. \tag{4.65}$$

The above reasoning leads us to expect approximately $\kappa|\Gamma_t|/\pi$ "non-decaying" singular values, which was the second heuristic we discussed. Applying the same reasoning to the adjoint of (4.65) gives us the corresponding value $\kappa \operatorname{diam} \Gamma_t$, which was the multipole heuristic. In regimes where we have an angle of interaction less than 2π, μ in (4.64) will be supported only on an arc in the circle, and the integration limits of (4.65) can be truncated appropriately. As a result, we derive the correct heuristic in these cases as well.

4.4 Efficient algorithms for building universal skeletons

Suppose that $\kappa_{\min}$, $\kappa_{\max}$, Γ_s, and Γ_t are given, and let $\mathbf{G}_\kappa \in \mathbb{C}^{m \times n}$ for any $\kappa > 0$ denote a discretization of the single, double, or combined layer operator that maps sources on Γ_s to targets on Γ_t with wavenumber κ. Given $\varepsilon > 0$ our goal is to construct an ID with interpolation matrix $\mathbf{U}$ and skeleton indices $\mathcal{I}$ such that

$$\|\mathbf{G}_\kappa - \mathbf{U}\mathbf{G}_\kappa(\mathcal{I}, :)\| < \varepsilon \tag{4.66}$$

for all $\kappa \in [\kappa_{\min}, \kappa_{\max}]$. In principle, we could directly discretize the restricted Fourier operator $\mathcal{F}_c$ with $c \geq \kappa_{\max}$ given by (4.61), but this would require discretizing an area integral in 2D and a volume integral in 3D and would also significantly overestimate the ranks in cases where the interactions between sources and targets are restricted to some angle. Discretizing $\mathcal{F}_c^* \mathcal{F}_c$ would remedy the former issue, but it neglects the latter and would also yield a loss of information in modes corresponding to singular values of $\mathcal{F}_c$ below the square root of the working precision.

In light of all this, we take the approach of Section 4.3.3 which constructed a universal basis by concatenating coefficient matrices with different wavenumbers. We again use the notation $\mathbf{G}^{(N)}_{[\kappa_{\min}, \kappa_{\max}]}$ to denote the matrix (4.38), where $\kappa_j^{(N)}$ still denotes the jth Chebyshev point of the second kind in $[\kappa_{\min}, \kappa_{\max}]$. For shorthand we will suppress the subscript and refer to $\mathbf{G}^{(N)}_{[\kappa_{\min}, \kappa_{\max}]}$ simply as $\mathbf{G}^{(N)}$. For all odd N we also introduce the notation

$$\widetilde{\mathbf{G}}^{(N)} = \begin{bmatrix} \mathbf{G}_{\kappa_2^{(N)}} & \mathbf{G}_{\kappa_4^{(N)}} & \cdots & \mathbf{G}_{\kappa_{N-1}^{(N)}} \end{bmatrix}, \tag{4.67}$$

so that $\widetilde{\mathbf{G}}^{(N)}$ is a submatrix of $\mathbf{G}^{(N)}$ that omits every other block.

The value of N needed depends non-trivially on the geometry and wavenumber interval, so in practice it is best determined adaptively. We start by setting the

parameter p equal to 1, assemble the matrix $\mathbf{G}^{(2^p+1)}$, and compute a matrix $\mathbf{U}^{(p)}$ and index vector $\mathcal{I}^{(p)}$ such that

$$\|\mathbf{G}^{(2^p+1)} - \mathbf{U}^{(p)}\mathbf{G}^{(2^p+1)}(\mathcal{I}^{(p)}, :)\| < \varepsilon. \tag{4.68}$$

This of course guarantees

$$\|\mathbf{G}_{\kappa_j^{(2^p+1)}} - \mathbf{U}^{(p)}\mathbf{G}_{\kappa_j^{(2^p+1)}}(\mathcal{I}^{(p)}, :)\| < \varepsilon, \quad j = 1, 2, \ldots, 2^p + 1. \tag{4.69}$$

In order to check that we have sampled at enough wavenumbers, we check the error on a finer grid of wavenumbers consisting of $\{\kappa_j^{(2^{p+1}+1)}\}_{j=1}^{2^{p+1}+1}$. Since $\{\kappa_j^{(2^p+1)}\}_{j=1}^{2^p+1} = \{\kappa_{2j+1}^{(2^{p+1}+1)}\}_{j=0}^{2^p}$, it suffices to simply check that

$$\left\|\mathbf{G}_{\kappa_{2j}^{(2^{p+1}+1)}} - \mathbf{U}^{(p)}\mathbf{G}_{\kappa_{2j}^{(2^{p+1}+1)}}(\mathcal{I}^{(p)}, :)\right\| < \varepsilon, \quad j = 1, 2, \ldots, 2^p. \tag{4.70}$$

If (4.70) is not met, then we increment the value of p by 1, build new $\mathbf{U}^{(p)}$ and $\mathcal{I}^{(p)}$ satisfying (4.69), and check the error (4.70). Note that in the process of evaluating the left-hand side of (4.70) in the previous iteration, we have already evaluated all of the entries of the matrix $\mathbf{G}^{(2^p+1)}$ in the current iteration. This may be looped with p incremented by 1 at each iteration until (4.70) is satisfied. This algorithm is summarized in Algorithm 4.1.

Algorithm 4.1 Build universal skeletons

Input: Wavenumber interval $[\kappa_{\min}, \kappa_{\max}]$, compression tolerance $\varepsilon > 0$, and a function handle that given $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ returns $\mathbf{G}_\kappa$

Output: Interpolation matrix $\mathbf{U}$ and skeletons indices $\mathcal{I}$

1: **for all** $p = 1, 2, \ldots$ **do**
2: Form $\mathbf{G}^{(2^p+1)}$ in (4.38)
3: Compute $\mathbf{U}^{(p)}$ and $\mathcal{I}^{(p)}$ such that (4.68) holds
4: **if** $\max_{j=1,2,\ldots,2^p} \left\|\mathbf{G}_{\kappa_{2j}^{(2^{p+1}+1)}} - \mathbf{U}^{(p)}\mathbf{G}_{\kappa_{2j}^{(2^{p+1}+1)}}(\mathcal{I}^{(p)}, :)\right\| < \varepsilon$ **then**
5: **return** $\mathbf{U} = \mathbf{U}^{(p)}$ and $\mathcal{I} = \mathcal{I}^{(p)}$
6: **end if**
7: **end for**

In practice we find that the termination criterion (4.70) is a good proxy for the quality of the universal skeletons throughout the entire interval $[\kappa_{\min}, \kappa_{\max}]$. Indeed, if (4.70) is met for some ε with $\mathbf{U}^{(p)}$ and $\mathcal{I}^{(p)}$ chosen from (4.69), we typically find that (4.66) also holds and almost never see the left-hand side of (4.66) exceeding 2ε. Furthermore, we observe that in the majority of cases only a few iterations are

required. In the numerical examples of the next section, typically only one to three iterations are required, and the number of iterations does not exceed six in any of the examples, despite considering intervals of wavenumbers corresponding to geometries differing in diameters by over a thousand wavelengths.

Still, it is rather unsatisfactory that (4.69) requires us to compute the ID of a matrix ($\mathbf{G}^{(2^p+1)}$) with a number of columns that grows exponentially with the iteration count. Fortunately, this can be accelerated by viewing the above procedure as a streaming algorithm. The key idea here is that instead of working with $\mathbf{G}^{(2^p+1)}$ at each iteration, we work primarily with a random sketch. In the first iteration, we compute

$$\mathbf{Y}^{(1)} = \mathbf{G}^{(3)}\mathbf{\Omega}^{(1)} \in \mathbb{C}^{m\times\widetilde{n}} \tag{4.71}$$

where $\mathbf{\Omega}^{(1)} \in \mathbb{C}^{3n\times\widetilde{n}}$ with $\widetilde{n} < 3n$ is either a normalized Gaussian or a sparse sign matrix (see [42, 51, 116, 154] and the references therein for discussion of sparse sign matrices). If $\widetilde{n}$ is chosen appropriately (in practice we find $\widetilde{n} \approx 2n$ is a good choice), then the leading left singular vectors of $\mathbf{Y}^{(1)}$ align well with those of $\mathbf{G}^{(3)}$ with high probability. We can therefore compute factors $\mathbf{U}^{(1)}$ and $\mathcal{I}^{(1)}$ such that (4.69) holds by instead computing an ID of the smaller matrix $\mathbf{Y}^{(1)}$. In subsequent iterations we can form a sketch of $\mathbf{G}^{(2^{p+1}+1)}$ given $\mathbf{Y}^{(p)} \in \mathbb{C}^{m\times k}$, a sketch of $\mathbf{G}^{(2^p+1)}$, by evaluating

$$\mathbf{Y}^{(p+1)} = \mathbf{Y}^{(p)} + \widetilde{\mathbf{G}}^{(2^{p+1}+1)}\mathbf{\Omega}^{(p+1)} \in \mathbb{C}^{m\times\widetilde{n}} \tag{4.72}$$

where $\mathbf{\Omega}^{(p+1)} \in \mathbb{C}^{2^p n\times\widetilde{n}}$ is another randomized embedding with columns drawn in the same fashion as those of $\mathbf{\Omega}^{(1)}$. When sparse sign matrices are used, (4.72) can be evaluated in $\mathcal{O}(m2^{p+1}n)$ floating point operations. Now $\mathbf{Y}^{(p+1)}$ may be used as a proxy for $\mathbf{G}^{(2^{p+1}+1)}$ in all subsequent steps of the algorithm, so the sizes of the matrices for which we must compute the ID remain constant. If the parameters are chosen appropriately, then there exist theoretical guarantees that state that using the sketches introduces negligible errors with high probability [51, 83, 115, 116, 161, 166]. Moreover, one can cheaply compute a "certificate of accuracy" using a small Gaussian test matrix at each iteration, which improves robustness [116, Section 12.2].

Algorithm 4.1 is rather suboptimal since at each iteration the new information added consists of entire coefficient matrices $\mathbf{G}_\kappa$ for varying values of κ, which we already know are of low numerical rank. Therefore, at each iteration a lot of redundant information is introduced. A faster algorithm may be obtained through randomized "coordinate-wise" sampling of the wavenumbers. In this case we fix a value of N such

that $N = 2^P + 1$ for some $P \in \mathbb{N}$, let $p = 1$, and set $\mathbf{A}^{(1)} = \sqrt{N/3}\mathbf{G}^{(3)} \in \mathbb{C}^{m \times 3n}$. An ID of $\mathbf{A}^{(p)}$ is computed, and $\mathbf{U}^{(p)}$ and $\mathcal{I}^{(p)}$ satisfying

$$\|\mathbf{A}^{(p)} - \mathbf{U}^{(p)}\mathbf{A}^{(p)}(\mathcal{I}^{(p)}, :)\| < \varepsilon \tag{4.73}$$

are produced. We now draw the matrix $\mathbf{B}^{(p)} \in \mathbb{C}^{m \times \widetilde{n}}$ (in practice we find that $\widetilde{n} \approx 2n$ works well) where the columns of $\mathbf{B}^{(p)}$ are chosen from the columns of $\mathbf{G}^{(N)}$ selected with uniform probability and without replacement, and scaled by a factor of $\sqrt{Nn/\widetilde{n}}$. In other words, $\mathbf{B}^{(p)}$ is given by

$$\mathbf{B}^{(p)} = \sqrt{\frac{Nn}{\widetilde{n}}} \begin{bmatrix} \mathbf{G}_{\kappa'_1}(:, i_1) & \mathbf{G}_{\kappa'_2}(:, i_2) & \dots & \mathbf{G}_{\kappa'_{\widetilde{n}}}(:, i_{\widetilde{n}}) \end{bmatrix} \tag{4.74}$$

where $\{i_j\}_{j=1}^{\widetilde{n}}$ is randomly drawn uniformly from $\{1, 2, \dots, Nn\}$ and $\{\kappa'_j\}_{j=1}^{\widetilde{n}}$ is randomly selected from $\{\kappa_j^{(N)}\}_{j=1}^{N}$. For our termination criterion, we check the condition

$$\max_{1 \le j \le \widetilde{n}} \left\| \mathbf{B}^{(p)}(:, j) - \mathbf{U}^{(p)}\mathbf{B}^{(p)}(\mathcal{I}^{(p)}, j) \right\| < \varepsilon. \tag{4.75}$$

If this condition is not satisfied, we increment p, and let

$$\mathbf{A}^{(p)} = \frac{1}{\sqrt{3n + p\widetilde{n}}} \begin{bmatrix} \sqrt{3n + (p-1)\widetilde{n}}\mathbf{A}^{(p-1)} & \sqrt{\widetilde{n}}\mathbf{B}^{(p-1)} \end{bmatrix}. \tag{4.76}$$

The process is then repeated with $\mathbf{U}^{(p)}$ and $\mathcal{I}^{(p)}$ produced by computing an ID of $\mathbf{A}^{(p)}$, a new matrix $\mathbf{B}^{(p)}$ drawn in an identical manner as $\mathbf{B}^{(1)}$, and the condition (4.75) checked. Again, this procedure is iterated until (4.75) is met, and sketching may be used to keep the size of $\mathbf{A}^{(p)}$ effectively constant across all iterations. This is summarized in Algorithm 4.2.

Since this algorithm depends on coordinate-wise sampling, its success is related to the concept of coherence in the matrix completion and randomized algorithm literature [33, 34, 116]. We are not aware of any results establishing coherence bounds on matrices like $\mathbf{G}^{(N)}$, and even if we had such a bound, the relevant theorems would inevitably be more pessimistic than what we observe. We find that this algorithm is robust in practice and convergence is usually achieved in merely a couple of iterations.

Applying these algorithms in recursive skeletonization is relatively straightforward. Given an interval $[\kappa_{\min}, \kappa_{\max}]$ and a tree structure, we simply compute universal skeletons for each block that must be compressed. After this precomputation, the compression stage for any wavenumber $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ at which we wish to do RS consists purely of assembling the sibling-, self-, and near- (if applicable) interactions. Since the ID has been used, this consists of forming submatrices of the coefficient matrix

Algorithm 4.2 Build universal skeletons (accelerated)

Input: Wavenumber interval $[\kappa_{\min}, \kappa_{\max}]$, compression tolerance $\varepsilon > 0$, and a function handle that given $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ returns $\mathbf{G}_\kappa$

Output: Interpolation matrix $\mathbf{U}$ and skeletons indices $\mathcal{I}$

$\mathbf{A}^{(1)} = \sqrt{N/3}\mathbf{G}^{(3)}$
for all $p = 1, 2, \ldots$ **do**
    Compute $\mathbf{U}^{(p)}$ and $\mathcal{I}^{(p)}$ such that (4.73) holds
    Form $\mathbf{B}^{(p)}$ with columns given by (4.74)
    if $\max_{j=1,2,\ldots,\widetilde{n}} \left\| \mathbf{B}^{(p)}(:, j) - \mathbf{U}^{(p)}\mathbf{B}^{(p)}(\mathcal{I}^{(p)}, j) \right\| < \varepsilon$ **then**
        return $\mathbf{U} = \mathbf{U}^{(p)}$ and $\mathcal{I} = \mathcal{I}^{(p)}$
    else
        Form $\mathbf{A}^{(p+1)}$ in (4.76)
    end if
end for

corresponding to wavenumber κ. We note that since each node in the hierarchical tree can be processed independently of the others, this step is highly parallelizable. After this any rank-structured matrix arithmetic (e.g., inversion or matrix-vector multiplication) can be done as normal. The trade-offs here are that the size of the discretization and the ranks in the rank-structured format are kept constant across $[\kappa_{\min}, \kappa_{\max}]$. In spite of this, we still achieve considerable accelerations in many regimes as illustrated in the next section. We refer to the variant of RS where universal skeletons are computed and used, as broadband recursive skeletonization or BRS for short. In the language of the FMM, the precomputation of the universal skeletons can be viewed as constructing multipole-to-multipole and local-to-local translation operators that are valid for all $\kappa \in [\kappa_{\min}, \kappa_{\max}]$. At each $\kappa \in [\kappa_{\min}, \kappa_{\max}]$, the multipole-to-local translation operators must, however, be computed. The trade-offs are now that the number of particles and the lengths of the multipole and local expansions must be kept constant across the interval of wavenumbers.

4.5 Applications and numerical experiments

We now illustrate BRS in various settings. All timings were done using a workstation with 2 Intel Xeon Gold 6254 processors (18 cores at 3.1GHz base frequency) with 768 GB of RAM, with all codes implemented in MATLAB 2019b. We use the FMMLIB2D [70] and FMM3D [63] libraries for 2D and 3D FMMs, respectively. The

zeta quadratures are computed using the codes at [167] and [168].[2]

4.5.1 2D boundary integral equation

For our first numerical example, we consider a standard combined field integral operator of the form (4.4) when the domain is the smooth, five-pointed star pictured in Fig. 4.10(a) and parametrized by $\boldsymbol{\gamma} = (\gamma_1, \gamma_2) : [0, 2\pi) \to \mathbb{R}^2$ where

$$\begin{aligned} \gamma_1(t) &= 1.5\cos(t) + 0.15\cos(6t) + 0.15\cos(4t) \\ \gamma_2(t) &= \sin(t) + 0.15\sin(6t) - 0.15\sin(4t). \end{aligned} \tag{4.77}$$

We will first study the performance and accuracy of BRS on intervals of wavenumbers $[1, \kappa_{\max}]$ for varying values of $\kappa_{\max}$. We will use 10^{th}-order zeta quadratures [170] and take the number of degrees of freedom in our discretization for each interval to be

$$400 + \left\lceil \frac{10|\Gamma_t|\kappa_{\max}}{2\pi} \right\rceil \approx 400 + \lceil 16.34\kappa_{\max} \rceil, \tag{4.78}$$

which corresponds to taking 400 plus 10 points times the length of the contour measured in wavelengths at the highest frequency. The values of $\kappa_{\max}$ range from 125 to 2000. The largest interval of wavenumbers, when $\kappa_{\max} = 2000$, corresponds to domains ranging from under a wavelength to over a thousand wavelengths in diameter. For each interval we first do a one-time computation of the universal skeletons. Then for each wavenumber, we obtain an HBS approximation to the coefficient matrix by simply evaluating the sibling-sibling and self-interaction matrices. After this we use standard algorithms for fast matrix-vector products, inversion, and application of the inverse. In our experiments we use a scattering matrix-based solver for inversion (see, for example Section 3.4.1 or [28]). Theoretically, in two dimensions these algorithms all scale linearly with the number of degrees of freedom, whereas inverting the rank-structured matrix scales cubically and matrix-vector products and applying the inverse scale quadratically with the wavenumber. These complexities all depend on the ranks of the various off-diagonal blocks which, since universal skeletons are used, stay constant across all wavenumbers. To gauge accuracy, we would like to measure the error

$$\frac{\|\mathbf{M}_\kappa - \mathbf{M}_\kappa^\varepsilon\|}{\|\mathbf{M}_\kappa\|} \tag{4.79}$$

where $\mathbf{M}_\kappa$ is the true coefficient matrix and $\mathbf{M}_\kappa^\varepsilon$ is the HBS approximation when the off-diagonal blocks are compressed to tolerance ε in the spectral norm. Since

[2]We are grateful to Bowei Wu for his help with using these codes and providing additional supplemental codes for working with the zeta quadratures.

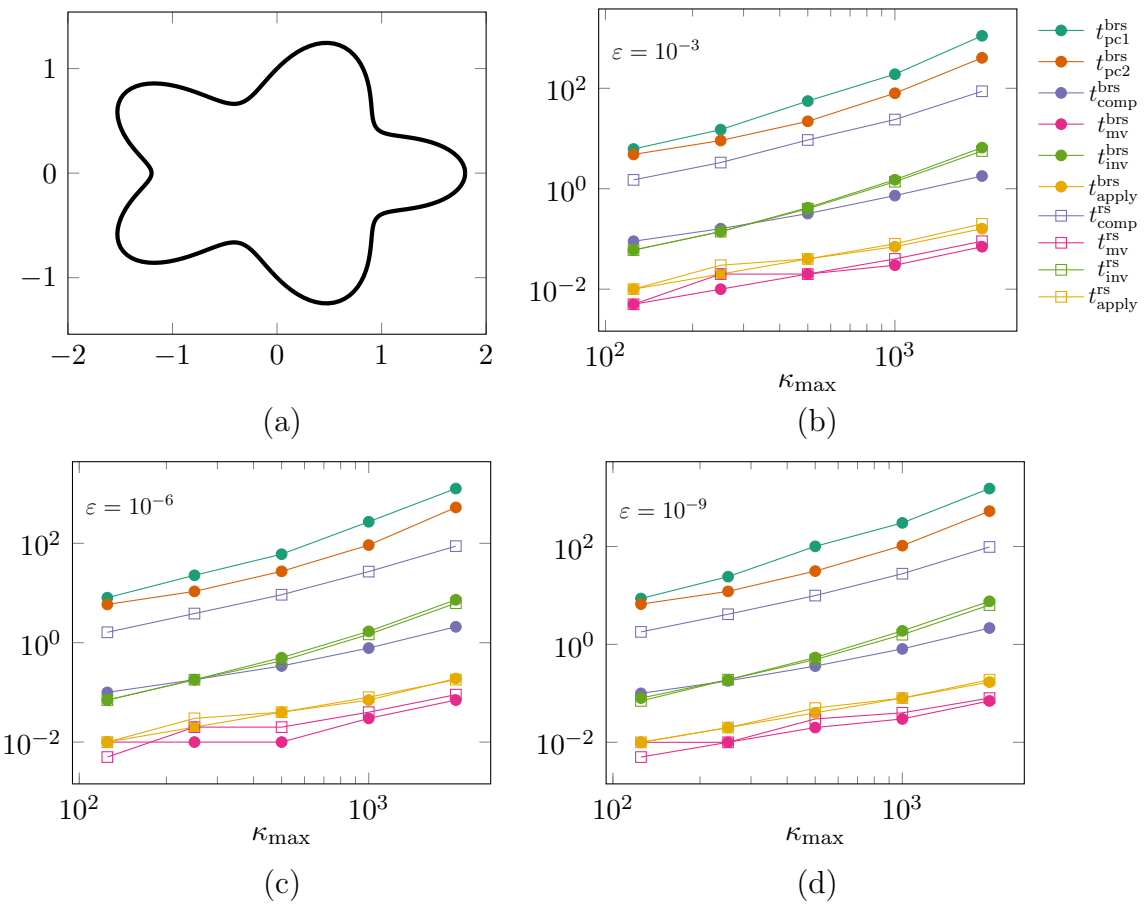


Figure 4.10: (a) Smooth, five-pointed star geometry as considered in Section 4.5.1. The arc length of the contour is about 10.27 and its diameter is about 3.39. (b–d) Timing data comparing rank-structured matrix algebra when BRS is used for an interval $[1, \kappa_{\max}]$ compared to when RS is naïvely applied at just $\kappa_{\max}$ when the off-diagonal blocks are compressed to tolerances 10^{-3}, 10^{-6}, and 10^{-9}, respectively.

computing (4.79) directly is prohibitively expensive, we instead use a power iteration where a high-accuracy FMM is used to compute matrix-vector products with $\mathbf{M}_\kappa$ and $\mathbf{M}_\kappa^*$. We approximate (4.79) in this fashion at $K = \lceil \kappa \rceil$ equispaced wavenumbers in the interval and take the maximum.

The following quantities are reported in Tables 4.2, 4.3, and 4.4, which consider the cases when the compression tolerance is 10^{-3}, 10^{-6}, and 10^{-9}, respectively:

$t^{\text{brs}}_{\text{pc1}}$	Time required for one-time computation of the universal skeletons for HBS matrices when Algorithm 4.1 is used
$t^{\text{brs}}_{\text{pc2}}$	Time required for one-time computation of the universal skeletons for HBS matrices when Algorithm 4.2 is used
$t^{\text{brs}}_{\text{comp}}$	Time required to obtain an HBS approximation to the coefficient matrix at each wavenumber after the universal skeletons are computed

$t_{\text{mv}}^{\text{brs}}$	Time required for a single matrix-vector product at each wavenumber with universal skeletons
$t_{\text{inv}}^{\text{brs}}$	Time required to build a rank-structured inverse at each wavenumber with universal skeletons
$t_{\text{apply}}^{\text{brs}}$	Time required to apply the inverse, after it is computed, at each wavenumber with universal skeletons
E_1^{brs}	Aforementioned error metric when Algorithm 4.1 is used to compute the universal skeletons
E_2^{brs}	Aforementioned error metric when Algorithm 4.2 is used to compute the universal skeletons

$[\kappa_{\min}, \kappa_{\max}]$	$t_{\text{pc1}}^{\text{brs}}$	$t_{\text{pc2}}^{\text{brs}}$	$t_{\text{comp}}^{\text{brs}}$	$t_{\text{mv}}^{\text{brs}}$	$t_{\text{inv}}^{\text{brs}}$	$t_{\text{apply}}^{\text{brs}}$	E_1^{brs}	E_2^{brs}
$[1, 125]$	6.18	4.82	0.09	0.00	0.06	0.01	$6.38 \cdot 10^{-4}$	$7.27 \cdot 10^{-4}$
$[1, 250]$	15.05	9.11	0.16	0.01	0.14	0.02	$1.30 \cdot 10^{-3}$	$1.73 \cdot 10^{-3}$
$[1, 500]$	55.71	22.00	0.32	0.02	0.42	0.04	$2.07 \cdot 10^{-3}$	$3.55 \cdot 10^{-3}$
$[1, 1000]$	190.94	79.78	0.73	0.03	1.51	0.07	$5.06 \cdot 10^{-3}$	$5.60 \cdot 10^{-3}$
$[1, 2000]$	1110.31	404.08	1.97	0.07	6.59	0.16	$9.48 \cdot 10^{-3}$	$2.40 \cdot 10^{-2}$

Table 4.2: Data for Section 4.5.1 when BRS is used and the compression tolerance $\varepsilon = 10^{-3}$

$[\kappa_{\min}, \kappa_{\max}]$	$t_{\text{pc1}}^{\text{brs}}$	$t_{\text{pc2}}^{\text{brs}}$	$t_{\text{comp}}^{\text{brs}}$	$t_{\text{mv}}^{\text{brs}}$	$t_{\text{inv}}^{\text{brs}}$	$t_{\text{apply}}^{\text{brs}}$	E_1^{brs}	E_2^{brs}
$[1, 125]$	8.04	5.89	0.10	0.01	0.07	0.01	$9.92 \cdot 10^{-7}$	$7.36 \cdot 10^{-7}$
$[1, 250]$	22.70	10.78	0.18	0.01	0.18	0.02	$1.42 \cdot 10^{-6}$	$2.23 \cdot 10^{-6}$
$[1, 500]$	60.37	27.28	0.34	0.01	0.50	0.04	$2.73 \cdot 10^{-6}$	$1.59 \cdot 10^{-6}$
$[1, 1000]$	271.37	92.12	0.78	0.03	1.69	0.07	$5.67 \cdot 10^{-6}$	$6.39 \cdot 10^{-6}$
$[1, 2000]$	1264.17	527.51	2.09	0.07	7.31	0.19	$8.36 \cdot 10^{-6}$	$1.10 \cdot 10^{-5}$

Table 4.3: Data for Section 4.5.1 when BRS is used and the compression tolerance $\varepsilon = 10^{-6}$

$[\kappa_{\min}, \kappa_{\max}]$	$t_{\text{pc1}}^{\text{brs}}$	$t_{\text{pc2}}^{\text{brs}}$	$t_{\text{comp}}^{\text{brs}}$	$t_{\text{mv}}^{\text{brs}}$	$t_{\text{inv}}^{\text{brs}}$	$t_{\text{apply}}^{\text{brs}}$	E_1^{brs}	E_2^{brs}
$[1, 125]$	8.65	6.66	0.10	0.01	0.08	0.01	$1.39 \cdot 10^{-9}$	$1.54 \cdot 10^{-9}$
$[1, 250]$	24.20	12.12	0.18	0.01	0.19	0.02	$3.24 \cdot 10^{-9}$	$1.98 \cdot 10^{-9}$
$[1, 500]$	100.62	31.44	0.36	0.02	0.54	0.04	$2.41 \cdot 10^{-9}$	$4.51 \cdot 10^{-9}$
$[1, 1000]$	303.56	103.99	0.81	0.03	1.89	0.08	$6.84 \cdot 10^{-9}$	$4.89 \cdot 10^{-9}$
$[1, 2000]$	1525.61	527.21	2.16	0.07	7.59	0.17	$6.70 \cdot 10^{-9}$	$4.20 \cdot 10^{-9}$

Table 4.4: Data for Section 4.5.1 when BRS is used and the compression tolerance $\varepsilon = 10^{-9}$

The data in these tables are summarized in Fig. 4.10(b–d). In all three cases we find roughly the same behavior. The computation of the universal skeletons is significantly more expensive than the other steps by roughly a couple orders of magnitude and using Algorithm 4.2 for this step yields a factor of 2–3 speedup over using Algorithm 4.1. Fortunately, this step must only be done once for the entire interval of wavenumbers. As $\kappa_{\max}$ doubles, the cost of each computation increases by a factor of 2–6, slightly better than the theoretical prediction of 8 for the computation of the universal skeletons and the inversion stage. Our error metric is also typically within an order of magnitude of the compression tolerance, which is standard. It is important to note that the accuracy lost is not due to our algorithms failing to obtain accurate universal skeletons but rather the accumulation of errors that would occur even at a single wavenumber (see Remark 3.8). This being the case, the error grows with the number of levels in the tree, which is why the errors increase with $\kappa_{\max}$. There also does not seem to be a substantial difference in accuracy between when Algorithm 4.1 and Algorithm 4.2 are used. In subsequent experiments we will only use Algorithm 4.2 since it is usually faster. The dependence of the timings on the compression tolerance ε is fairly weak with the cost of each computation growing substantially slower than $\log(1/\varepsilon)$.

To put these numbers into perspective, we compare them with the analogous values when regular RS is done at just the highest wavenumber. This is the focus of Tables 4.6, 4.7, and 4.8, which tabulate the quantities:

$t_{\text{comp}}^{\text{rs}}$	Time required to obtain an HBS approximation to the coefficient matrix without universal skeletons
$t_{\text{mv}}^{\text{rs}}$	Time required for a single matrix-vector product without universal skeletons
$t_{\text{inv}}^{\text{rs}}$	Time required to build a rank-structured inverse without universal skeletons
$t_{\text{apply}}^{\text{rs}}$	Time required to apply the inverse after it is computed without universal skeletons
E^{rs}	Aforementioned error metric without universal skeletons

κ	$t^{\text{rs}}_{\text{comp}}$	$t^{\text{rs}}_{\text{mv}}$	$t^{\text{rs}}_{\text{inv}}$	$t^{\text{rs}}_{\text{apply}}$	E^{rs}
125	1.50	0.00	0.06	0.01	$1.44 \cdot 10^{-3}$
250	3.32	0.02	0.14	0.03	$1.18 \cdot 10^{-3}$
500	9.35	0.02	0.40	0.04	$2.92 \cdot 10^{-3}$
1000	23.85	0.04	1.38	0.08	$2.89 \cdot 10^{-3}$
2000	87.29	0.09	5.64	0.20	$4.53 \cdot 10^{-3}$

Table 4.6: Data for Section 4.5.1 when regular RS compression is done at a single wavenumber and with compression tolerance $\varepsilon = 10^{-3}$

κ	$t^{\text{rs}}_{\text{comp}}$	$t^{\text{rs}}_{\text{mv}}$	$t^{\text{rs}}_{\text{inv}}$	$t^{\text{rs}}_{\text{apply}}$	E^{rs}
125	1.62	0.00	0.07	0.01	$2.73 \cdot 10^{-6}$
250	3.86	0.02	0.18	0.03	$2.19 \cdot 10^{-6}$
500	9.22	0.02	0.43	0.04	$9.06 \cdot 10^{-6}$
1000	26.94	0.04	1.47	0.08	$5.21 \cdot 10^{-6}$
2000	87.68	0.09	6.23	0.18	$1.05 \cdot 10^{-5}$

Table 4.7: Data for Section 4.5.1 when regular RS compression is done at a single wavenumber and with compression tolerance $\varepsilon = 10^{-6}$

κ	$t^{\text{rs}}_{\text{comp}}$	$t^{\text{rs}}_{\text{mv}}$	$t^{\text{rs}}_{\text{inv}}$	$t^{\text{rs}}_{\text{apply}}$	E^{rs}
125	1.80	0.00	0.07	0.01	$1.14 \cdot 10^{-9}$
250	4.13	0.01	0.19	0.02	$2.24 \cdot 10^{-9}$
500	9.93	0.03	0.49	0.05	$3.90 \cdot 10^{-9}$
1000	27.77	0.04	1.58	0.08	$4.29 \cdot 10^{-9}$
2000	97.74	0.08	6.37	0.19	$1.67 \cdot 10^{-8}$

Table 4.8: Data for Section 4.5.1 when regular RS compression is done at a single wavenumber and with compression tolerance $\varepsilon = 10^{-9}$

The data in these tables are also summarized in Fig. 4.10(b–d). We first examine the cost of inverting the HBS matrices. Across the board this is about 20% cheaper than with BRS. This is, of course, to be expected; the universal skeletons computed in BRS are valid at $\kappa_{\max}$, so the ranks when compression is done at just $\kappa_{\max}$ are necessarily lower and therefore we should expect inversion to be faster. The same reasoning applies to the cost of matrix-vector products and applying the inverse, but since these computations are light on arithmetic, simply consisting of a series of matrix-vector products, communication costs are more pronounced. As a result, these costs are observed to be essentially the same as in BRS.

It is important to compare the compression cost in regular RS with the corresponding costs in BRS. In all of these experiments, the compression cost in regular RS is substantially faster than the cost of computing the universal skeletons for BRS. Even when the accelerated algorithm is used to compute the universal skeletons, this difference ranges from a factor of 4–6. However, if the universal skeletons have already been computed, then forming an HBS approximation to the coefficient matrix using BRS is about a factor of 10–40 faster than the cost of computing it from scratch. Since the compression cost dominates the inversion cost, this suggests that BRS can yield a significant speedup in the per wavenumber cost in multi-frequency applications.

Of course, when regular RS is used at each wavenumber the rank-structured matrix algebra costs depend on the wavenumber, and this dependence is theoretically quadratic or cubic. This is in contrast to BRS where these costs are effectively constant for an entire interval. To gain a practical speedup, the constants in the complexities are important. We will more closely study this trade-off by focusing on the case when $\kappa_{\max} = 1000$ and examining how the costs associated with BRS (displayed in the fourth row of Tables 4.2, 4.3, and 4.4) compare to the corresponding *average costs* of dense linear algebra and regular RS in various subintervals of $[1, 1000]$. For this the size of the discretization should be chosen based on the wavenumber, so for $\kappa \in [1, 1000]$ we will always take $400 + \lceil 16.34\kappa \rceil$ degrees of freedom. In Tables 4.10 and 4.11, we tabulate the following quantities:

$\overline{t_{\text{form}}^{\text{dense}}}$	Average time required to assemble the full dense coefficient matrix for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$
$\overline{t_{\text{mv}}^{\text{dense}}}$	Average time required for a single matrix-vector product after dense coefficient matrix is formed for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$
$\overline{t_{\text{LU}}^{\text{dense}}}$	Average time required to compute a dense LU factorization of the coefficient matrix for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$
$\overline{t_{\text{apply}}^{\text{dense}}}$	Average time required to do a single backward and forward substitution after the dense LU factors have been computed for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$
$\overline{t_{\text{comp}}^{\text{rs}}}$	Average time required to compute an HBS approximation of the coefficient matrix for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ without universal skeletons
$\overline{t_{\text{mv}}^{\text{rs}}}$	Average time required to do a single matrix-vector product with the HBS matrix for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ without universal skeletons
$\overline{t_{\text{inv}}^{\text{rs}}}$	Average time required to build an inverse of an HBS matrix for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$
$\overline{t_{\text{apply}}^{\text{rs}}}$	Average time required to apply the inverse, after it is computed, for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ without universal skeletons

	dense				RS ($\varepsilon = 10^{-3}$)			
$[\kappa_{\min}, \kappa_{\max}]$	$\overline{t_{\text{form}}^{\text{dense}}}$	$\overline{t_{\text{mv}}^{\text{dense}}}$	$\overline{t_{\text{LU}}^{\text{dense}}}$	$\overline{t_{\text{apply}}^{\text{dense}}}$	$\overline{t_{\text{comp}}^{\text{rs}}}$	$\overline{t_{\text{mv}}^{\text{rs}}}$	$\overline{t_{\text{inv}}^{\text{rs}}}$	$\overline{t_{\text{apply}}^{\text{rs}}}$
$[1, 1000]$	3.84	0.02	4.41	0.32	9.71	0.01	0.52	0.03
$[500, 1000]$	6.36	0.04	6.71	0.50	14.96	0.02	0.82	0.05
$[750, 1000]$	9.35	0.04	10.30	0.70	17.71	0.03	1.07	0.06
$[875, 1000]$	10.31	0.06	12.18	0.78	19.07	0.03	1.17	0.06
$[999, 1000]$	11.17	0.06	14.24	0.80	23.73	0.03	1.38	0.06

Table 4.10: Data for Section 4.5.1 showing average times required by dense linear algebra and regular RS ($\varepsilon = 10^{-3}$) are done for various intervals of wavenumbers

	RS ($\varepsilon = 10^{-6}$)				RS ($\varepsilon = 10^{-9}$)			
$[\kappa_{\min}, \kappa_{\max}]$	$\overline{t_{\text{comp}}^{\text{rs}}}$	$\overline{t_{\text{mv}}^{\text{rs}}}$	$\overline{t_{\text{inv}}^{\text{rs}}}$	$\overline{t_{\text{apply}}^{\text{rs}}}$	$\overline{t_{\text{comp}}^{\text{rs}}}$	$\overline{t_{\text{mv}}^{\text{rs}}}$	$\overline{t_{\text{inv}}^{\text{rs}}}$	$\overline{t_{\text{apply}}^{\text{rs}}}$
$[1, 1000]$	10.28	0.02	0.56	0.03	10.50	0.02	0.62	0.04
$[500, 1000]$	15.70	0.02	0.90	0.05	16.20	0.02	0.98	0.05
$[750, 1000]$	18.88	0.03	1.18	0.07	19.43	0.03	1.27	0.07
$[875, 1000]$	20.74	0.03	1.33	0.07	22.61	0.03	1.45	0.07
$[999, 1000]$	24.18	0.03	1.48	0.07	27.57	0.03	1.61	0.08

Table 4.11: Data for Section 4.5.1 showing average times required by regular RS ($\varepsilon = 10^{-6}$ and $\varepsilon = 10^{-9}$) are done for various intervals of wavenumbers

To interpret this data, let us consider a couple of concrete cases. Suppose we are interested in solving a single integral equation at K different wavenumbers in the interval $[1, 1000]$ where the wavenumbers are chosen fairly uniformly. If we were to use dense linear algebra, then we would expect the total cost to be approximately K times $\overline{t_{\text{form}}^{\text{dense}}} + \overline{t_{\text{LU}}^{\text{dense}}} + \overline{t_{\text{LU}}^{\text{dense}}} \approx 8.57$ s. On the other hand, if we were to do regular RS with $\varepsilon = 10^{-9}$ at each iteration, then we would expect the cost to be approximately K times $\overline{t_{\text{comp}}^{\text{rs}}} + \overline{t_{\text{inv}}^{\text{rs}}} + \overline{t_{\text{apply}}^{\text{rs}}} \approx 11.16$ s. From this it follows that in this case, there would be no point in doing regular RS over dense linear algebra. For BRS with $\varepsilon = 10^{-9}$ the total cost would be the time it takes to compute the universal skeletons (103.99 s) plus K times $t_{\text{comp}}^{\text{brs}} + t_{\text{inv}}^{\text{brs}} + t_{\text{apply}}^{\text{brs}} \approx 2.78$ s. Per wavenumber, BRS is about 3 times faster than dense linear algebra and 4 times faster than regular RS, and for sufficiently large K (in this case, $K \geq 18$), BRS yields a lower total running time. In cases where we restrict ourselves to a narrower interval around $\kappa_{\max} = 1000$ or where the wavenumbers are drawn more densely at higher wavenumbers, the advantages of BRS grow more pronounced. For example, if we were to stay in the same setting but instead were to consider the wavenumber interval $[999, 1000]$, the average per wavenumber costs of

dense linear algebra and regular RS would change to 26.21 s and 29.26 s, respectively, but the per wavenumber cost associated with BRS would essentially remain the same. The average per wavenumber speedup with BRS would then be a factor of about 10 when compared with both methods and the break-even point for the universal skeletons computation would be at about $K = 5$.

Remark 4.5. While the above analysis has shown that the strategy of applying BRS for a large interval of wavenumbers such as $[1, 1000]$ can still yield practical acceleration, this is a rather crude strategy. Better practical performance could be achieved by partitioning $[\kappa_{\min}, \kappa_{\max}]$ into a series of subintervals and then applying BRS to each interval. This would lower the time required to compute the universal skeletons as well. How this partitioning should be done is very application dependent, so we do not explore this matter further here.

4.5.2 2D eigenvalue problems

Closely related to scattering problems is the problem of finding the eigenvalues of the Laplacian. One approach to this is to formulate the problem as a standard boundary integral equation for an interior Helmholtz equation and solve the resulting nonlinear eigenvalue problem to determine the wavenumbers at which the integral operator is singular. These wavenumbers are called the *eigenfrequencies* and are the square roots of the eigenvalues. In this section we will consider the problem of finding the eigenfrequencies in given intervals $[\kappa_{\min}, \kappa_{\max}]$ of the drum domain introduced in [176], pictured in Fig. 4.11(a), and whose boundary is parametrized by $r(\theta) = 1 + 0.2\cos 3\theta + 0.3\sin 2\theta$ for $\theta \in [0, 2\pi)$.

We shall accomplish this by finding κ such that the operator $\mathcal{A}_\kappa : L^2(\Gamma) \to L^2(\Gamma)$ given by

$$[\mathcal{A}_\kappa \sigma](\boldsymbol{x}) = \frac{1}{2}\sigma(\boldsymbol{x}) - \int_\Gamma \frac{\partial G_\kappa}{\partial n_{\boldsymbol{y}}}(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y})\, ds(\boldsymbol{y}) \tag{4.80}$$

is singular. There exist many different methods of solving this nonlinear eigenvalue problem, and here we will take one of the simplest approaches, which was used in [113, Section 6.3]. To this end, let $\mathbf{M}_\kappa$ denote a discretization of $\mathcal{A}_\kappa$ in the manner described in Section 4.2.1. We use 16^{th}-order zeta quadratures [170], and for a given interval $[\kappa_{\min}, \kappa_{\max}]$, we will take the number of points in our discretization to be $\lceil 10|\Gamma|\kappa_{\max}/2\pi \rceil \approx 11.60\kappa_{\max}$. We will find the eigenfrequencies by finding the values of κ where $\mathbf{M}_\kappa$ becomes singular. This in turn will be accomplished by root-finding on the function $f(\kappa) = \sigma_{\min}(\mathbf{M}_\kappa)$.

When $\kappa_{\max}$ is small, it is efficient to simply form the dense matrix $\mathbf{M}_\kappa$ and to compute its smallest singular value using bidiagonalization and the QR algorithm. For larger values of $\kappa_{\max}$, it can be faster to utilize direct solver techniques. We shall do this by computing an HBS approximation to $\mathbf{M}_\kappa$ for each wavenumber, inverting it using the HBS-LU/RSF algorithm [89, 112], and then using a power iteration. Naturally, BRS can be used to accelerate this first step. We apply dense linear algebra, regular RS, and BRS to the problem of finding the eigenfrequencies of the drum in the intervals $[500, 500.1]$ and $[1000, 1000.1]$. For regular RS and BRS, we set the compression tolerance to 10^{-10}. The root-finding is done by initially evaluating f on a coarse grid in $[\kappa_{\min}, \kappa_{\max}]$ and finding local minima, followed by a refinement step, which uses a quasi-Newton algorithm as implemented in MATLAB's `fminunc` algorithm. The resulting solver finds all eigenfrequencies in our two intervals to six correct digits after the decimal in about 72 function evaluations per root. The performance of the various methods is summarized in Table 4.13, which tabulates the following quantities:

$\overline{t_{\text{sv}}^{\text{brs}}}$	Average time required to compute the smallest singular value by forming an HBS matrix using the universal skeletons, inverting the rank-structured matrix using an HBS-LU factorization, and then a power iteration in the interval
$\overline{t_{\text{sv}}^{\text{rs}}}$	Average time required to compute the smallest singular value by forming an HBS matrix from scratch at each wavenumber, inverting the rank-structured matrix using an HBS-LU factorization, and then a power iteration in the interval
$\overline{t_{\text{sv}}^{\text{dense}}}$	Average time required to compute the smallest singular value by forming the dense coefficient matrix, and then bidiagonalization and QR in the interval

	$t_{\text{pc2}}^{\text{brs}}$	$\overline{t_{\text{sv}}^{\text{brs}}}$	$\overline{t_{\text{sv}}^{\text{rs}}}$	$\overline{t_{\text{sv}}^{\text{dense}}}$
$[500, 500.1]$ (26 eigenvalues)	12.35	1.35	7.10	8.64
$[1000, 1000.1]$ (52 eigenvalues)	30.33	3.01	18.39	42.13

Table 4.13: Data for Section 4.5.2

We see that each evaluation of f when BRS is used is about a factor of 5 on the interval $[500, 500.1]$ and a factor of 6 on the interval $[1000, 1000.1]$ faster than when regular RS is used, which is in turn faster than dense linear algebra. Also, the one-time cost of computing the universal skeletons is negligible in light of the hundreds

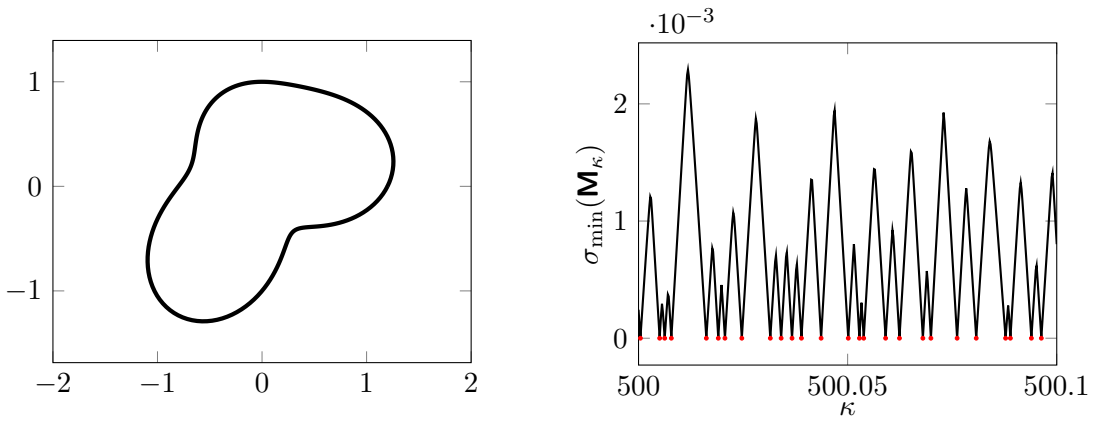


Figure 4.11: (a) Drum domain. The diameter of the domain is about 2.69 and the arc length of its boundary is approximately 7.29. (b) Plot of $\sigma_{\min}(\mathbf{M}_\kappa)$ when $\kappa \in [500, 500.1]$. The red dots denote the computed eigenfrequencies.

of function evaluations we are doing. The eigenfrequencies and a plot of the function f on the interval $[500, 500.1]$ are shown in Table 4.13(b).

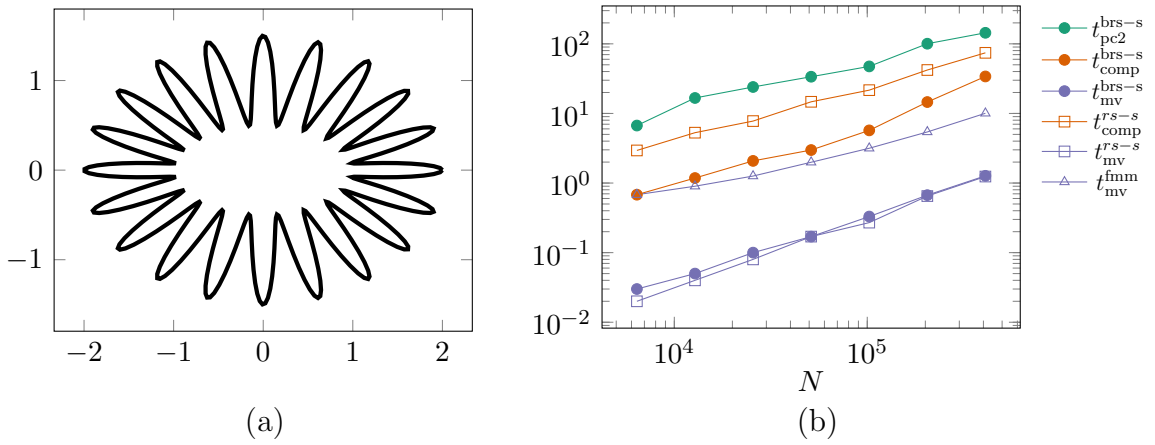
Remark 4.6. The optimization strategy employed here to solve the nonlinear eigenvalue problem is rather primitive. A better approach would be to follow the approach developed in [176] and extended in [10] and compute the Fredholm determinant of $2\mathcal{A}_\kappa$ where $\mathcal{A}_\kappa$ is given in (4.80). This is the subject of ongoing work.

4.5.3 2D fast matrix-vector products

We now consider the problem of rapidly applying the single layer operator in (4.26) when Γ_s and Γ_t are both taken to be the same contour Γ given by the smooth, 20-pointed star contour in Fig. 4.12(a) and parametrized by $\boldsymbol{\gamma} = (\gamma_1, \gamma_2) : [0, 2\pi) \to \mathbb{R}^2$ where

$$\begin{aligned} \gamma_1(t) &= 0.375\cos(t) + 0.0375\cos(21t) + 0.0375\cos(19t) \\ \gamma_2(t) &= 0.25\sin(t) + 0.0375\sin(21t) - 0.0375\sin(19t). \end{aligned} \tag{4.81}$$

This operator is discretized as usual, but unlike the experiments in the previous section, we now use a strongly-admissible $\mathcal{H}^2$-matrix format and therefore use a strongly-admissible version of recursive skeletonization [120]. We will study how BRS, regular RS, and the FMM compare when we are interested in matrix-vector multiplications with discretizations of the single layer operator of varying sizes when $\kappa \in [1, 2000]$. The compression tolerance for BRS and regular RS is taken to be 10^{-10}; the precision flag in the FMMLIB2D library was set to 4. Since we have



(a) (b)

Figure 4.12: (a) Smooth, 20-pointed star contour. The arc length of the contour is approximately 10.33 and its diameter is exactly 1. (b) Timing data for the experiments in Section 4.5.3

not yet implemented parallelized versions of our solvers, we disable OpenMP in the FMM. We believe that this is a fair comparison since the $\mathcal{H}^2$-matrix matrix-vector product algorithm we use has essentially the same data flow as the FMM, so we expect the algorithms to have the same level of parallelization potential. We also use a rather simplistic approach and build the $\mathcal{H}^2$-matrices based on uniform quad-trees on $[-0.5, 0.5] \times [-0.5, 0.5]$. An adaptive subdivision of space would obviously be better. For our error metric in this section, we use

$$\max_{i=1,2,\ldots,10} \frac{\|\widehat{\mathbf{g}}_i^{(\kappa)} - \mathbf{M}_\kappa \mathbf{g}_i^{(\kappa)}\|}{\|\mathbf{M}_\kappa \mathbf{g}_i^{(\kappa)}\|} \tag{4.82}$$

where $\{\mathbf{g}_i^{(\kappa)}\}_{i=1}^{10}$ are random vectors with Gaussian entries and $\widehat{\mathbf{g}}_i^{(\kappa)}$ is the computed matrix-vector product. We check (4.82) for several different wavenumbers and take the maximum. The product $\mathbf{M}_\kappa \mathbf{g}_i^{(\kappa)}$ is evaluated using a higher precision FMM; specifically, we use the precision flag 6 in FMMLIB2D.

The following quantities are reported in Table 4.15 for varying numbers of degrees of freedom:

N	Total number of degrees of freedom
$t_{\text{pc2}}^{\text{brs-s}}$	Time required for one-time computation of universal skeletons for $\mathcal{H}^2$-matrices when Algorithm 4.2 is used
$t_{\text{comp}}^{\text{brs-s}}$	Time required to obtain an $\mathcal{H}^2$-matrix representation of the coefficient matrix at each wavenumber after the computation of the universal skeletons
$t_{\text{mv}}^{\text{brs-s}}$	Time required for a single matrix-vector product at each wavenumber with universal skeletons

$\overline{t_{\text{comp}}^{\text{rs-s}}}$	Average time to compute an $\mathcal{H}^2$-matrix approximation to the coefficient matrix for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ without universal skeletons
$\overline{t_{\text{mv}}^{\text{rs-s}}}$	Average time to do a single matrix-vector product with the $\mathcal{H}^2$-matrix for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$ without universal skeletons
$\overline{t_{\text{mv}}^{\text{fmm}}}$	Average time to do a single matrix-vector product using the FMM for $\kappa \in [\kappa_{\min}, \kappa_{\max}]$
$\widehat{E}^{\text{brs-s}}$	Aforementioned error metric for BRS
$\widehat{E}^{\text{rs-s}}$	Aforementioned error metric for regular RS
$\widehat{E}^{\text{fmm}}$	Aforementioned error metric for FMM

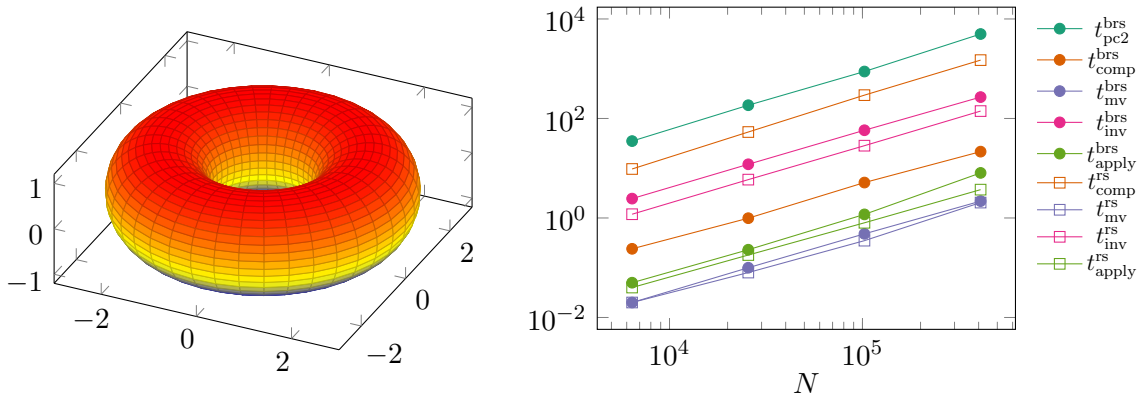
N	$t_{\text{pc2}}^{\text{brs-s}}$	$t_{\text{comp}}^{\text{brs-s}}$	$t_{\text{mv}}^{\text{brs-s}}$	$\widehat{E}^{\text{brs-s}}$	$\overline{t_{\text{comp}}^{\text{rs-s}}}$	$\overline{t_{\text{mv}}^{\text{rs-s}}}$	$\widehat{E}^{\text{rs-s}}$	$\overline{t_{\text{mv}}^{\text{fmm}}}$	$\widehat{E}^{\text{fmm}}$
6400	6.70	0.68	0.03	$6.28 \cdot 10^{-12}$	2.94	0.02	$2.57 \cdot 10^{-10}$	0.68	$1.04 \cdot 10^{-16}$
12800	16.68	1.18	0.05	$2.29 \cdot 10^{-11}$	5.29	0.04	$3.59 \cdot 10^{-10}$	0.90	$8.44 \cdot 10^{-17}$
25600	23.99	2.08	0.10	$3.18 \cdot 10^{-11}$	7.77	0.08	$5.76 \cdot 10^{-10}$	1.26	$6.32 \cdot 10^{-17}$
51200	33.69	2.98	0.17	$4.42 \cdot 10^{-11}$	14.70	0.17	$9.62 \cdot 10^{-10}$	1.99	$5.31 \cdot 10^{-17}$
102400	47.39	5.71	0.33	$7.31 \cdot 10^{-11}$	21.61	0.27	$2.01 \cdot 10^{-9}$	3.16	$4.99 \cdot 10^{-17}$
204800	100.43	14.58	0.67	$1.04 \cdot 10^{-10}$	42.07	0.65	$2.69 \cdot 10^{-9}$	5.42	$2.98 \cdot 10^{-17}$
409600	144.09	34.00	1.27	$2.38 \cdot 10^{-10}$	74.28	1.24	$5.16 \cdot 10^{-9}$	10.05	$2.89 \cdot 10^{-17}$

Table 4.15: Data for Section 4.5.3

We note that the time required to compute the universal skeletons is about 2–4 times greater than the average time required to compute an $\mathcal{H}^2$-matrix approximation from scratch at each wavenumber. The latter is in turn 2–5 times slower than forming the $\mathcal{H}^2$-matrix at each wavenumber after the universal skeletons have been computed. This factor would be even larger using a more sophisticated wavenumber partitioning strategy with BRS, as mentioned in Remark 4.5. The cost of each matrix-vector multiply using BRS is only slightly greater than that with regular RS. Both are about a factor of 10–20 times faster than applying the FMM at each wavenumber. We expect to do even better than this with an implementation using a fully adaptive quad-tree. As for accuracy, we see that the matrix-vector products computed using the FMM are almost identical to the reference values, despite the latter being computed with a higher precision flag. The error metrics for BRS and regular RS are in line with what we would expect.

4.5.4 3D boundary integral equation

Our last example will be a simple extension of the experiments in Section 4.5.1 to three dimensions. We consider a torus pictured in Fig. 4.13(a) and which can be



(a) (b)

Figure 4.13: (a) Torus geometry. The diameter of the domain is about 6. (b) Timing data for Section 4.5.4

parametrized by $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3) : [0, 2\pi) \times [0, 2\pi) \to \mathbb{R}^3$ where

$$\begin{aligned} \gamma_1(\theta, \phi) &= (2 + \cos(\theta)) \cos(\phi) \\ \gamma_2(\theta, \phi) &= (2 + \cos(\theta)) \sin(\phi) \\ \gamma_2(\theta, \phi) &= \sin(\theta). \end{aligned} \tag{4.83}$$

Note that the torus is about 6 units in diameter. We use 3^{rd}-order zeta quadratures [169] for simplicity since these are diagonal, but the timings would be similar for higher-order quadratures. We will compare BRS with regular RS as the number of degrees of freedom increases when HBS matrices are used with compression tolerance $\varepsilon = 10^{-5}$, $\kappa_{\min} = 1$, and $\kappa_{\max} = 10$. The timings for BRS are shown in Table 4.16 and regular RS in Table 4.17, and the timings are illustrated in Fig. 4.13(b).

N	$t^{\text{brs}}_{\text{pc2}}$	$t^{\text{brs}}_{\text{comp}}$	$t^{\text{brs}}_{\text{mv}}$	$t^{\text{brs}}_{\text{inv}}$	$t^{\text{brs}}_{\text{apply}}$	E^{brs}
6400	35.10	0.24	0.02	2.45	0.05	$7.77 \cdot 10^{-6}$
25600	184.64	0.99	0.10	12.00	0.23	$1.38 \cdot 10^{-5}$
102400	877.95	5.15	0.48	57.98	1.18	$3.21 \cdot 10^{-5}$
409600	4980.86	21.54	2.18	268.97	8.05	$2.00 \cdot 10^{-3}$

Table 4.16: Data for Section 4.5.4 when BRS is used

N	$\overline{t^{\text{rs}}_{\text{comp}}}$	$\overline{t^{\text{rs}}_{\text{mv}}}$	$\overline{t^{\text{rs}}_{\text{inv}}}$	$\overline{t^{\text{rs}}_{\text{apply}}}$	E^{rs}
6400	9.65	0.02	1.19	0.04	$3.53 \cdot 10^{-5}$
25600	53.40	0.08	5.90	0.18	$1.98 \cdot 10^{-4}$
102400	294.41	0.35	28.37	0.80	$1.10 \cdot 10^{-3}$
409600	1489.03	2.04	140.83	3.71	$5.33 \cdot 10^{-3}$

Table 4.17: Data for Section 4.5.4 when regular RS is used

We see that the compression and inversion costs of both regular RS and BRS are slightly better than the theoretical asymptotic $\mathcal{O}(N^{3/2})$ complexity. In every case the cost of regular RS compression is about a factor of 4 less than the cost of computing the universal skeletons in BRS. However, after the universal skeletons have been computed, the cost of the compression step at each wavenumber in BRS is about 40–70 times cheaper than that of regular RS. Given the generally higher ranks in 3D, we see that now there is a substantial increase in the subsequent matrix algebra costs in BRS over regular RS. Inversion, for example, is consistently about a factor of 2 slower. Nonetheless, this trade-off will still yield practical advantages to using BRS in a broad range of applications. Restricting ourselves to the case where there are 102,400 degrees of freedom, the cost per wavenumber to do a single compression, inversion, and application of the inverse using BRS is approximately $t^{\text{brs}}_{\text{comp}} + t^{\text{brs}}_{\text{inv}} + t^{\text{brs}}_{\text{apply}} \approx 64.30$ s. The analogous cost for regular RS is $t^{\text{rs}}_{\text{comp}} + t^{\text{rs}}_{\text{inv}} + t^{\text{rs}}_{\text{apply}} \approx 323.58$ s, which is about a factor of 5 greater. Moreover, the cost of computing the universal skeletons is compensated for after only about 4 wavenumbers.

Chapter 5

Conclusions

In this thesis we have considered the numerical solution of Helmholtz equations. Chapter 2 introduced a new method for achieving high-order convergence for problems with corner singularities in two dimensions. In Chapter 3 we introduced a new direct solver for the Lippmann–Schwinger integral equation formulation of inhomogeneous scattering problems in the plane. Lastly, in Chapter 4 we introduced a new technique for accelerating existing direct solvers in broadband environments.

In closing we outline some directions for future work, a few of which are the topics of ongoing investigations. The material in Chapter 2 has many extensions. As aforementioned, rigorous theory for the Helmholtz solver is lacking, and the approach in Section 2.5 perhaps warrants further investigation. Furthermore, we are unsure of the best way to stabilize our solver in the Helmholtz case (see Remark 2.4). For this it could be interesting to couple a boundary integral formulation for the smooth part of the solution with sums of dipoles to resolve singularities. It is also currently unknown whether or not our approach can be extended to problems in three dimensions; we expect cases where the domain has edges to be especially challenging. Of special interest to us is accelerating the solution of the linear least-squares problem in the solver. This is the dominant cost, and the coefficient matrices have similar rank structure to those arising in boundary integral equation discretizations, which could be exploited. There have been a few recent algorithms for fast solvers in this setting (e.g., [90, 110, 171]), but we believe that there is still more to do in this area. Moreover, advances here could also be applied to integral equation formulations of quasiperiodic and multilayered problems [16, 40, 41, 175].

The complexity of the solver introduced in Chapter 3 is $\mathcal{O}(N^{3/2})$ in 2D and increases to $\mathcal{O}(N^2)$ in 3D. Naturally, it is desirable to reduce these costs to linear or quasilinear complexity as mentioned in Remark 3.9. Some preliminary numerical experiments suggest that the hierarchical interpolative factorization technique introduced in [89]

may be especially well-suited for this purpose. We are currently exploring this further. Of course, how to develop solvers that are subcubic in the wavenumber (see Remark 3.10) for generic scattering potentials is a major open problem. We are also in the process of extending the ideas here to develop solvers for some simple 3D domains, such as rough surfaces and tori, using zeta quadratures [169]. A preprint on this topic should soon be available.

There exist a plethora of directions in which the work in Chapter 4 can be extended. Ideally, the heuristic arguments presented in Section 4.3 would be developed into a rigorous theory. Furthermore, we believe that there is significant room for improvement in the algorithms in Section 4.4 and hope to develop faster algorithms with provable guarantees in the future. As mentioned in Remark 4.5, it also remains to develop adaptive algorithms that partition an interval of wavenumbers into subintervals, each with its own set of universal skeletons. We are also actively developing new solvers that utilize broadband recursive skeletonization for applications such as eigenvalue and resonance finding, as mentioned in Remark 4.6, and solving the time-dependent wave equation. More on this will be reported at a later date.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, volume 55. US Government Printing Office, 1964.

[2] B. K. Alpert. Hybrid Gauss-trapezoidal quadrature rules. *SIAM Journal on Scientific Computing*, 20(5):1551–1584, 1999.

[3] S. Ambikasaran, C. Borges, L.-M. Imbert-Gerard, and L. Greengard. Fast, adaptive, high-order accurate discretization of the Lippmann–Schwinger equation in two dimensions. *SIAM Journal on Scientific Computing*, 38(3):A1770–A1787, 2016.

[4] S. Ambikasaran and E. Darve. An $\mathcal{O}(N \log N)$ fast direct solver for partial hierarchically semi-separable matrices. *Journal of Scientific Computing*, 57(3):477–501, 2013.

[5] S. Ambikasaran and E. Darve. The inverse fast multipole method. *arXiv preprint arXiv:1407.1572*, 2014.

[6] H. Ammari, G. Bao, and A. W. Wood. An integral equation method for the electromagnetic scattering from cavities. *Mathematical Methods in the Applied Sciences*, 23(12):1057–1072, 2000.

[7] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, et al. *LAPACK Users' Guide*. SIAM, 1999.

[8] T. G. Anderson, O. P. Bruno, and M. Lyon. High-order, dispersionless "fast-hybrid" wave equation solver. Part I: $\mathcal{O}(1)$ sampling cost via incident-field windowing and recentering. *SIAM Journal on Scientific Computing*, 42(2):A1348–A1379, 2020.

[9] P. Anselone. *Collectively Compact Operator Approximation Theory*. Prentice-Hall Series in Automatic Computation. Prentice-Hall Inc., 1971.

[10] T. Askham and M. Rachh. A boundary integral equation approach to computing eigenvalues of the Stokes operator. *Advances in Computational Mathematics*, 46(2):1–42, 2020.

[11] K. E. Atkinson. *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1997.

[12] I. Babuška and S. Sauter. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Journal on Numerical Analysis*, 34(6):2392–2423, 1997.

[13] J. Ballani and D. Kressner. Matrices with hierarchical low-rank structures. In *Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications*, pages 161–209. Springer, 2016.

[14] A. H. Barnett and T. Betcke. Stability and convergence of the method of fundamental solutions for Helmholtz problems on analytic domains. *Journal of Computational Physics*, 227(14):7003–7026, 2008.

[15] A. H. Barnett and T. Betcke. An exponentially convergent nonpolynomial finite element method for time-harmonic scattering from polygons. *SIAM Journal on Scientific Computing*, 32(3):1417–1441, 2010.

[16] A. H. Barnett, G. R. Marple, S. Veerapaneni, and L. Zhao. A unified integral equation scheme for doubly-periodic Laplace and Stokes boundary value problems in two dimensions. *Communications on Pure and Applied Mathematics*, 71(11):2334–2380, 2018.

[17] M. Bebendorf. *Hierarchical Matrices*. Springer–Verlag Berlin Heidelberg, 2008.

[18] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185–200, 1994.

[19] S. Bernstein. Sur la meilleure approximation de $|x|$ par des polynomes de degrés donnés. *Acta Mathematica*, 37(1):1–57, 1914.

[20] T. Betcke. A GSVD formulation of a domain decomposition method for planar eigenvalue problems. *IMA Journal of Numerical Analysis*, 27(3):451–478, 2007.

[21] G. Beylkin, C. Kurcz, and L. Monzón. Grids and transforms for band-limited functions in a disk. *Inverse Problems*, 23(5):2059, 2007.

[22] C. Borges, A. Gillman, and L. Greengard. High resolution inverse scattering in two dimensions using recursive linearization. *SIAM Journal on Imaging Sciences*, 10(2):641–664, 2017.

[23] S. Börm. $\mathcal{H}^2$-matrix arithmetics in linear complexity. *Computing*, 77(1):1–28, 2006.

[24] S. Börm. Construction of data-sparse $\mathcal{H}^2$-matrices by hierarchical compression. *SIAM Journal on Scientific Computing*, 31(3):1820–1839, 2009.

[25] S. Börm. *Efficient Numerical Methods for Non-local Operators: $\mathcal{H}^2$-matrix Compression, Algorithms and Analysis*, volume 14 of *Tracts in Mathematics*. European Mathematical Society, 2010.

[26] J. P. Boyd. Approximation of an analytic function on a finite real interval by a bandlimited function and conjectures on properties of prolate spheroidal functions. *Applied and Computational Harmonic Analysis*, 15(2):168–176, 2003.

[27] J. Bremer. A quasilinear complexity algorithm for the numerical simulation of scattering from a two-dimensional radially symmetric potential. *Journal of Computational Physics*, page 109401, 2020.

[28] J. Bremer, A. Gillman, and P.-G. Martinsson. A high-order accurate accelerated direct solver for acoustic scattering from surfaces. *BIT Numerical Mathematics*, 55(2):367–397, 2015.

[29] J. Bremer and Z. Gimbutas. On the numerical evaluation of the singular integrals of scattering theory. *Journal of Computational Physics*, 251:327–343, 2013.

[30] P. D. Brubeck, Y. Nakatsukasa, and L. N. Trefethen. Vandermonde with Arnoldi. *SIAM Review*, 63(2):405–415, 2021.

[31] O. M. Bucci and G. Franceschetti. On the spatial bandwidth of scattered fields. *IEEE Transactions on Antennas and Propagation*, 35(12):1445–1455, 1987.

[32] O. M. Bucci and G. Franceschetti. On the degrees of freedom of scattered fields. *IEEE Transactions on Antennas and Propagation*, 37(7):918–926, 1989.

[33] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[34] E. J. Candès and T. Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[35] S. Chandrasekaran, P. Dewilde, M. Gu, W. Lyons, and T. Pals. A fast solver for HSS representations via sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1):67–81, 2006.

[36] S. Chandrasekaran, M. Gu, and T. Pals. A fast ULV decomposition solver for hierarchically semiseparable representations. *SIAM Journal on Matrix Analysis and Applications*, 28(3):603–622, 2006.

[37] Y. Chen. A fast, direct algorithm for the Lippmann–Schwinger integral equation in two dimensions. *Advances in Computational Mathematics*, 16(2):175–190, 2002.

[38] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin. On the compression of low rank matrices. *SIAM Journal on Scientific Computing*, 26(4):1389–1404, 2005.

[39] W. C. Chew, J.-M. Jin, and E. Michielssen. *Fast and Efficient Algorithms in Computational Electromagnetics*. Artech House, 2001.

[40] M. H. Cho. Spectrally-accurate numerical method for acoustic scattering from doubly-periodic 3D multilayered media. *Journal of Computational Physics*, 393:46–58, 2019.

[41] M. H. Cho and A. H. Barnett. Robust fast direct integral equation solver for quasi-periodic scattering problems with a large number of layers. *Optics express*, 23(2):1775–1799, 2015.

[42] M. B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 278–287. SIAM, 2016.

[43] D. Colton and R. Kress. *Integral Equation Methods in Scattering Theory*. Classics in Applied Mathematics. SIAM, 2013.

[44] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*, volume 93. Springer–Verlag New York, 4$^{\text{th}}$ edition, 2019.

[45] E. Corona, P.-G. Martinsson, and D. Zorin. An $\mathcal{O}(N)$ direct solver for integral equations on the plane. *Applied and Computational Harmonic Analysis*, 38(2):284–317, 2015.

[46] M. Costabel. On the spectrum of volume integral operators in acoustic scattering. In *Integral Methods in Science and Engineering*, pages 119–127. Springer, 2015.

[47] P. Coulier, H. Pouransari, and E. Darve. The inverse fast multipole method: using a fast approximate direct solver as a preconditioner for dense linear systems. *SIAM Journal on Scientific Computing*, 39(3):A761–A796, 2017.

[48] W. Crutchfield, Z. Gimbutas, L. Greengard, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao. Remarks on the implementation of wideband FMM for the Helmholtz equation in two dimensions. *Contemporary Mathematics*, 408(01), 2006.

[49] *NIST Digital Library of Mathematical Functions*. `http://dlmf.nist.gov/`, Release 1.1.0 of 2020-12-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[50] A. Doicu, Y. Eremin, and T. Wriedt. *Acoustic and Electromagnetic Scattering Analysis Using Discrete Sources*. Academic Press, 2000.

[51] Y. Dong and P.-G. Martinsson. Simpler is better: a comparative study of randomized algorithms for computing the CUR decomposition. *arXiv preprint arXiv:2104.05877*, 2021.

[52] R. Duan and V. Rokhlin. High-order quadratures for the solution of scattering problems in two dimensions. *Journal of Computational Physics*, 228(6):2152–2174, 2009.

[53] S. C. Eisenstat. On the rate of convergence of the Bergman–Vekua method for the numerical solution of elliptic boundary value problems. *SIAM Journal on Numerical Analysis*, 11(3):654–680, 1974.

[54] B. Engquist and A. Majda. Absorbing boundary conditions for numerical simulation of waves. *Proceedings of the National Academy of Sciences*, 74(5):1765–1766, 1977.

[55] B. Engquist and L. Ying. Fast directional multilevel algorithms for oscillatory kernels. *SIAM Journal on Scientific Computing*, 29(4):1710–1737, 2007.

[56] B. Engquist, L. Ying, et al. A fast directional algorithm for high frequency acoustic scattering in two dimensions. *Communications in Mathematical Sciences*, 7(2):327–345, 2009.

[57] B. Engquist and H. Zhao. Approximate separability of the Green's function of the Helmholtz equation in the high frequency limit. *Communications on Pure and Applied Mathematics*, 71(11):2220–2274, 2018.

[58] O. G. Ernst and M. J. Gander. Why it is difficult to solve Helmholtz problems with classical iterative methods. In *Numerical Analysis of Multiscale Problems*, pages 325–363. Springer, 2012.

[59] G. Fairweather and A. Karageorghis. The method of fundamental solutions for elliptic boundary value problems. *Advances in Computational Mathematics*, 9(1-2):69, 1998.

[60] D. Gaier. *Lectures on Complex Approximation*, volume 188. Birkhäuser Basel, 1987.

[61] A. Gillman, A. Barnett, and P.-G. Martinsson. A spectrally accurate direct solution technique for frequency-domain scattering problems with variable media. *BIT Numerical Mathematics*, 55(1):141–170, 2015.

[62] A. Gillman, P. Young, and P.-G. Martinsson. A direct solver with $\mathcal{O}(N)$ complexity for integral equations on one-dimensional domains. *Frontiers of Mathematics in China*, 7(2):217–247, 2012.

[63] Z. Gimbutas, L. Greengard, L. Lu, D. M. Jeremy Magland, M. O'Neil, M. Rachh, and V. Rokhlin. FMM3D. `https://github.com/flatironinstitute/FMM3D`. Release 1.0.0.

[64] A. Gončar. On the rapidity of rational approximation of continuous functions with characteristic singularities. *Matematicheskii Sbornik*, 2(4):561, 1967.

[65] A. Gopal and P.-G. Martinsson. An accelerated, high-order accurate direct solver for the Lippmann–Schwinger equation for acoustic scattering in the plane. *arXiv preprint arXiv:2007.12718*, 2020.

[66] A. Gopal and L. N. Trefethen. New Laplace and Helmholtz solvers. *Proceedings of the National Academy of Sciences*, 116(21):10223–10225, 2019.

[67] A. Gopal and L. N. Trefethen. Representation of conformal maps by rational functions. *Numerische Mathematik*, 142(2):359–382, 2019.

[68] A. Gopal and L. N. Trefethen. Solving Laplace problems with corner singularities via rational functions. *SIAM Journal on Numerical Analysis*, 57(5):2074–2094, 2019.

[69] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[70] L. Greengard and Z. Gimbutas. FMMLIB2D. `https://github.com/zgimbutas/fmmlib2d`. Version 1.2.3, released March 2021.

[71] L. Greengard, D. Gueyffier, P.-G. Martinsson, and V. Rokhlin. Fast direct solvers for integral equations in complex three-dimensional domains. *Acta Numerica*, pages 1–33, 2009.

[72] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.

[73] L. Greengard and V. Rokhlin. On the numerical solution of two-point boundary value problems. *Communications on Pure and Applied Mathematics*, 44(4):419–452, 1991.

[74] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.

[75] P. Greengard. *On Generalized Prolate Spheroidal Functions*. PhD thesis, Yale University, 2019.

[76] P. Greengard and K. Serkh. On generalized prolate spheroidal functions. *arXiv preprint arXiv:1811.02733*, 2018.

[77] P. Grisvard. *Elliptic Problems in Nonsmooth Domains*. SIAM, 2011.

[78] M. Gu and S. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.

[79] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[80] W. Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*, volume 49. Springer–Verlag Berlin Heidelberg, 2015.

[81] W. Hackbusch, B. Khoromskij, and S. A. Sauter. On $\mathcal{H}^2$-matrices. In *Lectures on Applied Mathematics*, pages 9–29. Springer, Berlin, Heidelberg, 2000.

[82] W. Hackbusch and B. N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic. Part II: application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.

[83] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[84] S. Hao, A. H. Barnett, P.-G. Martinsson, and P. Young. High-order accurate methods for Nyström discretization of integral equations on smooth curves in the plane. *Advances in Computational Mathematics*, 40(1):245–272, 2014.

[85] J. Helsing and R. Ojala. Corner singularities for elliptic problems: integral equations, graded meshes, quadrature, and compressed inverse preconditioning. *Journal of Computational Physics*, 227(20):8820–8840, 2008.

[86] P. Henrici. A survey of I. N. Vekua's theory of elliptic partial differential equations with analytic coefficients. *Zeitschrift für Angewandte Mathematik und Physik*, 8(3):169–203, 1957.

[87] P. Henrici. *Applied and Computational Complex Analysis*, volume 3 of *Wiley Classics Library*. John Wiley & Sons, 1993.

[88] K. Ho and L. Greengard. A fast direct solver for structured linear systems by recursive skeletonization. *SIAM Journal on Scientific Computing*, 34(5):A2507–A2532, 2012.

[89] K. Ho and L. Ying. Hierarchical interpolative factorization for elliptic operators: integral equations. *Communications on Pure and Applied Mathematics*, 69:1314–1353, 2016.

[90] K. L. Ho and L. Greengard. A fast semi-direct least squares algorithm for hierarchically block separable matrices. *SIAM Journal on Matrix Analysis and Applications*, 35(2):725–748, 2014.

[91] A. Hochman, Y. Leviatan, and J. K. White. On the use of rational-function fitting methods for the solution of 2D Laplace boundary-value problems. *Journal of Computational Physics*, 238:337–358, 2013.

[92] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

[93] J. G. Hoskins and V. Rokhlin. A fast adaptive algorithm for scattering from a two-dimensional radially-symmetric potential. *Yale Technical Report TR-1538*, 2019.

[94] J. G. Hoskins, V. Rokhlin, and K. Serkh. On the numerical solution of elliptic partial differential equations on polygonal domains. *SIAM Journal on Scientific Computing*, 41(4):A2552–A2578, 2019.

[95] S. Kapur and V. Rokhlin. High-order corrected trapezoidal quadrature rules for singular functions. *SIAM Journal on Numerical Analysis*, 34(4):1331–1356, 1997.

[96] M. Katsurada. A mathematical study of the charge simulation method II. *Journal of the Faculty of Science, the University of Tokyo, Section IA, Mathematics*, 36(1):135–162, 1989.

[97] M. Katsurada. Charge simulation method using exterior mapping functions. *Japan Journal of Industrial and Applied Mathematics*, 11(1):47, 1994.

[98] M. Katsurada and H. Okamoto. A mathematical study of the charge simulation method I. *Journal of the Faculty of Science, the University of Tokyo, Section IA, Mathematics*, 35(3):507–518, 1988.

[99] M. Köhl and S. Rjasanow. Multifrequency analysis for the Helmholtz equation. *Computational Mechanics*, 32(4):234–239, 2003.

[100] P. Kolm and V. Rokhlin. Numerical quadratures for singular and hypersingular integrals. *Computers & Mathematics with Applications*, 41(3-4):327–352, 2001.

[101] W. Y. Kong, J. Bremer, and V. Rokhlin. An adaptive fast direct solver for boundary integral equations in two dimensions. *Applied and Computational Harmonic Analysis*, 31(3):346–369, 2011.

[102] N. S. Koshlyakov, M. M. Smirnov, and E. B. Gliner. *Differential Equations of Mathematical Physics*. North Holland Publishing Company, 1964.

[103] R. Kress. *Linear Integral Equations*, volume 82 of *Applied Mathematical Sciences*. Springer–Verlag New York, 3rd edition, 2014.

[104] J. Lai, S. Ambikasaran, and L. Greengard. A fast direct solver for high frequency scattering from a large cavity in two dimensions. *SIAM Journal on Scientific Computing*, 36(6):B887–B903, 2014.

[105] J. Lai, L. Greengard, and M. O'Neil. Robust integral formulations for electromagnetic scattering from three-dimensional cavities. *Journal of Computational Physics*, 345:1–16, 2017.

[106] H. J. Landau and H. O. Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty II. *Bell System Technical Journal*, 40(1):65–84, 1961.

[107] H. J. Landau and H. O. Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty III: the dimension of the space of essentially time- and band-limited signals. *Bell System Technical Journal*, 41(4):1295–1336, 1962.

[108] R. R. Lederman. Numerical algorithms for the computation of generalized prolate spheroidal functions. *arXiv preprint arXiv:1710.02874*, 2017.

[109] R. S. Lehman. Developments at an analytic corner of solutions of elliptic partial differential equations. *Journal of Mathematics and Mechanics*, pages 727–760, 1959.

[110] Y. Liu. *The numerical solution of frequency-domain acoustic and electromagnetic periodic scattering problems*. PhD thesis, Dartmouth College, 2016.

[111] G. G. Lorentz, M. v. Golitschek, and Y. Makovoz. *Constructive Approximation: Advanced Problems*, volume 304. Springer–Verlag Berlin Heidelberg, 1996.

[112] P.-G. Martinsson. *Fast Direct Solvers for Elliptic PDEs*, volume 96 of *CBMS–NSF conference series*. SIAM, 2019.

[113] P.-G. Martinsson and V. Rokhlin. A fast direct solver for boundary integral equations in two dimensions. *Journal of Computational Physics*, 205(1):1–23, 2005.

[114] P.-G. Martinsson and V. Rokhlin. A fast direct solver for scattering problems involving elongated structures. *Journal of Computational Physics*, 221(1):288–302, 2007.

[115] P.-G. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30(1):47–68, 2011.

[116] P.-G. Martinsson and J. A. Tropp. Randomized numerical linear algebra: foundations and algorithms. *Acta Numerica*, 29:403–572, 2020.

[117] J. M. Melenk. *On generalized finite element methods*. PhD thesis, University of Maryland at College Park, 1995.

[118] J. M. Melenk. Operator adapted spectral element methods I: harmonic and generalized harmonic polynomials. *Numerische Mathematik*, 84(1):35–69, 1999.

[119] E. Michielssen, A. Boag, and W. C. Chew. Scattering from elongated objects: direct solution in $O(N \log^2 N)$ operations. *IEE Proceedings – Microwaves, Antennas, and Propagation*, 143(4):277 – 283, 1996.

[120] V. Minden, K. L. Ho, A. Damle, and L. Ying. A recursive skeletonization factorization based on strong admissibility. *Multiscale Modeling & Simulation*, 15(2):768–796, 2017.

[121] A. Moiola, R. Hiptmair, and I. Perugia. Plane wave approximation of homogeneous Helmholtz solutions. *Zeitschrift für angewandte Mathematik und Physik*, 62(5):809, 2011.

[122] A. Moiola, R. Hiptmair, and I. Perugia. Vekua theory for the Helmholtz operator. *Zeitschrift für Angewandte Mathematik und Physik*, 62(5):779, 2011.

[123] N. E. Myridis and C. Chamzas. Sampling on concentric circles [MRI image reconstruction application]. *IEEE Transactions on Medical Imaging*, 17(2):294–299, 1998.

[124] D. J. Newman. Rational approximation to $|x|$. *The Michigan Mathematical Journal*, 11(1):11–14, 1964.

[125] S. Ohnuki and W. C. Chew. Truncation error analysis of multipole expansion. *SIAM Journal on Scientific Computing*, 25(4):1293–1306, 2004.

[126] A. Osipov, V. Rokhlin, and H. Xiao. *Prolate Spheroidal Wave Functions of Order Zero*, volume 187 of *Applied Mathematical Sciences*. Springer, Boston, MA, 2013.

[127] M. O'Neil, L. Greengard, and A. Pataki. On the efficient representation of the half-space impedance Green's function for the Helmholtz equation. *Wave Motion*, 51(1):1–13, 2014.

[128] D. P. Petersen and D. Middleton. Sampling and reconstruction of wave-number-limited functions in N-dimensional Euclidean spaces. *Information and Control*, 5(4):279–323, 1962.

[129] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):187–207, 1985.

[130] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *Journal of Computational Physics*, 86(2):414–439, 1990.

[131] V. Rokhlin. Sparse diagonal forms for translation operators for the Helmholtz equation in two dimensions. *Applied and Computational Harmonic Analysis*, 5(1):36–67, 1998.

[132] N. Schryer. Constructive approximation of solutions to linear elliptic boundary value problems. *SIAM Journal on Numerical Analysis*, 9(4):546–572, 1972.

[133] C. Schwab. *p-and hp-finite Element Methods: Theory and Applications in Solid and Fluid Mechanics*. Oxford University Press, 1998.

[134] K. Serkh and V. Rokhlin. On the solution of elliptic partial differential equations on regions with corners. *Journal of Computational Physics*, 305:150–171, 2016.

[135] K. Serkh and V. Rokhlin. On the solution of the Helmholtz equation on regions with corners. *Proceedings of the National Academy of Sciences*, 113(33):9171–9176, 2016.

[136] Y. Shkolnisky. Prolate spheroidal wave functions on a disc——integration and approximation of two-dimensional bandlimited functions. *Applied and Computational Harmonic Analysis*, 22(2):235–256, 2007.

[137] Y. Shkolnisky, M. Tygert, and V. Rokhlin. Approximation of bandlimited functions. *Applied and Computational Harmonic Analysis*, 21(3):413–420, 2006.

[138] D. Slepian. Prolate spheroidal wave functions, Fourier analysis and uncertainty IV: extensions to many dimensions; generalized prolate spheroidal functions. *Bell System Technical Journal*, 43(6):3009–3057, 1964.

[139] D. Slepian. Some comments on Fourier analysis, uncertainty and modeling. *SIAM Review*, 25(3):379–393, 1983.

[140] D. Slepian and H. O. Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty I. *Bell System Technical Journal*, 40(1):43–63, 1961.

[141] J. Song and W. C. Chew. Error analysis for the truncation of multipole expansion of vector Green's functions. *IEEE Microwave and Wireless Components Letters*, 11(7):311–313, 2001.

[142] H. Stahl. Best uniform rational approximation of $|x|$ on $[-1, 1]$. *Matematicheskii Sbornik*, 183(8):85–118, 1993.

[143] H. Stahl. Best uniform rational approximation of x^α on $[0, 1]$. *Bulletin of the American Mathematical Society*, 28(1):116–122, 1993.

[144] P. Starr and V. Rokhlin. On the numerical solution of two-point boundary value problems II. *Communications on Pure and Applied Mathematics*, 47(8):1117–1159, 1994.

[145] F. Stenger. Explicit nearly optimal linear rational approximation with preassigned poles. *Mathematics of Computation*, 47(175):225–252, 1986.

[146] G. Still. On density and approximation properties of special solutions of the Helmohltz equation. *Zeitschrift für Angewandte Mathematik und Mechanik*, 72(7):277–290, 1992.

[147] M. Stolper. Computing and compression of the boundary element matrices for the Helmholtz equation. *Journal of Numerical Mathematics*, 2004.

[148] M. Stolper and S. Rjasanow. A compression method for the Helmholtz equation. In *Numerical Mathematics and Advanced Applications*, pages 786–795. Springer, 2004.

[149] G. Toraldo Di Francia. Degrees of freedom of an image. *Journal of the Optical Society of America*, 59(7):799–804, 1969.

[150] L. N. Trefethen. 8-digit Laplace solutions on polygons? Posting on NA Digest at `http://www.netlib.org/na-digest-html/18/v18n48.html` on 3 December 2018.

[151] L. N. Trefethen. Lightning Laplace solver. `https://people.maths.ox.ac.uk/trefethen/laplace.m`. Version 6, released March 2020.

[152] L. N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, 2019.

[153] L. N. Trefethen, Y. Nakatsukasa, and J. Weideman. Exponential node clustering at singularities for rational approximation, quadrature, and PDEs. *Numerische Mathematik*, pages 1–28, 2020.

[154] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. *Advances in Neural Information Processing Systems*, 30, 2017.

[155] J. Unnikrishnan and M. Vetterli. Sampling high-dimensional bandlimited fields on low-dimensional manifolds. *IEEE Transactions on Information Theory*, 59(4):2103–2127, 2012.

[156] R. S. Varga, A. Ruttan, and A. J. Carpenter. Numerical results on best uniform rational approximation of $|x|$ on $[-1, 1]$. *Matematicheskii Sbornik*, 74(2):271, 1993.

[157] I. N. Vekua. *New Methods for Solving Elliptic Equations*. North-Holland Series in Applied Mathematics & Mechanics. North Holland Publishing Company, 1967.

[158] F. Vico, L. Greengard, and M. Ferrando. Fast convolution with free-space Green's functions. *Journal of Computational Physics*, 323:191–203, 2016.

[159] B. Vioreanu and V. Rokhlin. Spectra of multiplication operators as a numerical tool. *SIAM Journal on Scientific Computing*, 36(1):A267–A288, 2014.

[160] O. Von Estorff, S. Rjasanow, M. Stolper, and O. Zaleski. Two efficient methods for a multifrequency solution of the Helmholtz equation. *Computing and Visualization in Science*, 8(3-4):159–167, 2005.

[161] S. Voronin and P.-G. Martinsson. Efficient algorithms for CUR and interpolative matrix decompositions. *Advances in Computational Mathematics*, 43(3):495–516, 2017.

[162] P. R. Wallace. *Mathematical Analysis of Physical Problems (Revised Edition)*. Dover Publications, 2011.

[163] J. L. Walsh. *Interpolation and Approximation by Rational Functions in the Complex Domain*, volume 20 of *Colloquium Publications*. American Mathematical Society, 1935.

[164] A. Willers. The Helmholtz equation in disturbed half-spaces. *Mathematical Methods in the Applied Sciences*, 9(1):312–323, 1987.

[165] W. J. Wiscombe. Improved Mie scattering algorithms. *Applied Optics*, 19(9):1505–1509, 1980.

[166] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335–366, 2008.

[167] B. Wu. ZetaTrap2D. `https://github.com/bobbielf2/ZetaTrap2D`.

[168] B. Wu. ZetaTrap3D. `https://github.com/bobbielf2/ZetaTrap3D`.

[169] B. Wu and P.-G. Martinsson. Corrected trapezoidal rules for boundary integral equations in three dimensions. *arXiv preprint arXiv:2007.02512*, 2020.

[170] B. Wu and P.-G. Martinsson. Zeta correction: a new approach to constructing corrected trapezoidal quadrature rules for singular integral operators. *arXiv preprint arXiv:2007.13898*, 2020.

[171] Y. Xi, J. Xia, S. Cauley, and V. Balakrishnan. Superfast and stable structured solvers for Toeplitz least squares via randomized sampling. *SIAM Journal on Matrix Analysis and Applications*, 35(1):44–72, 2014.

[172] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li. Fast algorithms for hierarchically semiseparable matrices. *Numerical Linear Algebra with Applications*, 17(6):953–976, 2010.

[173] H. Xiao and Z. Gimbutas. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications*, 59(2):663–676, 2010.

[174] H. Zhang and B. Zhang. A novel integral equation for scattering by locally rough surfaces and application to the inverse problem. *SIAM Journal on Applied Mathematics*, 73(5):1811–1829, 2013.

[175] Y. Zhang and A. Gillman. A fast direct solver for two dimensional quasi-periodic multilayered media scattering problems. *BIT Numerical Mathematics*, 61(1):141–171, 2021.

[176] L. Zhao and A. Barnett. Robust and efficient solution of the drum problem via Nyström approximation of the Fredholm determinant. *SIAM Journal on Numerical Analysis*, 53(4):1984–2007, 2015.