ABSTRACT

NORTH, EVAN ISAAC. Non-Iterative Domain Decomposition using the Method of Difference Potentials. (Under the direction of Semyon Tsynkov.)

The Helmholtz equation governs the propagation of time-harmonic waves. For domains many wavelengths in size, it becomes intractable to solve directly, even on modern computers. Non-overlapping Domain Decomposition Methods (DDMs) attempt to alleviate the cost growth by partitioning the domain into smaller, simpler subdomains thus creating subproblems that are coupled to one another along their interfaces. In this dissertation, we describe a framework that utilizes the method of difference potentials (MDP) to resolve the said coupling and enables the simultaneous solution of the subproblems.

Traditionally, DDMs resolve this coupling by an iterative process that alternates between directly solving a localized approximation of the subproblem and updating the resulting boundary conditions using parameterized transmission conditions. These iterations are repeated until convergence, and the rate of convergence relies heavily on the chosen transmission conditions. In contrast to traditional DDMs our method is non-iterative, utilizing only a single global step for the boundary and interface data followed by the solution of the decoupled subproblems.

The MDP reduces the Helmholtz equation on each subdomain to a Calderon's boundary equation with projection on its boundary. The unknowns for the Calderon's equation are the Dirichlet and Neumann data. Coupling between neighboring subdomains is rendered by applying their respective Calderon's equations to the same data at the common interface. Solutions on individual subdomains are computed concurrently using a straightforward direct solver.

The Calderon's operators are expensive to compute but can be stored and reused for similar problems. In the context of domain decomposition, this property extends to subproblems that share common properties — such as wavenumber or domain shape — so if several subproblems across the decomposition are similar, their corresponding operators only need to be computed once. Domains with properties that are periodically repeated are a natural choice for applying this methodology. The efficacy of the method is demonstrated by simulating wave propagation within a photonic crystal ring resonator (PCRR), a device used in micro-scale electronics for filtering particular frequencies from a signal.

The domain for a PCRR simulation can be constructed by starting with a lattice of evenly-spaced scattering rods. Rods are then removed in particular patterns so that the absence of rods leaves the impression of a traditional ring resonator. The resulting structure of a PCRR domain suggests a decomposition into two unique blocks: a square centered over a scattering rod and a square without a scattering rod. By building a PCRR from these "building blocks", one can quickly and easily redesign a new prototype for simulation because the pre-computation of the Calderon's operators only needs done once for each kind of building block.

The scope of this dissertation covers the construction, implementation, and numerical simula-

tions of the two fundamental subdomains mentioned above: a square with a constant wavenumber and a square whose piece-wise constant wavenumber is defined to be split over a simple shape, such as a circle, within the domain. These two subdomains form the building blocks that general rectangular domains can be broken into. Although we restrict ourselves to these simple cases for convenience, many of the formulas and ideas translate directly to more sophisticated examples.

Non-Iterative Domain Decomposition using the Method of Difference Potentials

by
Evan Isaac North

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2022

APPROVED BY:

Alina Chertock	Zhilin Li
Alen Alexanderian	Eli Turkel External Member

Semyon Tsynkov
Chair of Advisory Committee

BIOGRAPHY

Evan Isaac North was raised in the small town of North Baltimore in northwest Ohio. He attended the Honors Tutorial College at Ohio University where he earned a B.S. in Mathematics and jump-started the program's Math Club and Putnam exam group. The summer before his senior year, Evan participated in a Research Experience for Undergraduates through the Mathematical Biosciences Institute. It was here that he met his future wife and decided to apply to graduate school. In 2016, Evan began the Applied Mathematics Ph.D. program at North Carolina State University, earning his Master of Science in Applied mathematics in 2018. During his time at NCSU, Evan helped run the math department's student-led mentoring program, Undergrads Union Grads. In the fall of 2020, Evan transitioned from teaching to a research assistantship under the direction of Dr. Semyon Tsynkov, which resulted in the work of this dissertation.

ACKNOWLEDGEMENTS

I would like to thank the following people, each of whom played their own unique role in supporting me throughout graduate school and aiding the completion of this dissertation:

God, my creator, for creating a world so beautiful and rich with secrets to be discovered, and for walking alongside me every step of the way.

Dr. Semyon Tsynkov, for his incredible patience and help along the way. He has always been willing to explain things as many times as it takes without making me feel foolish for continuing to ask. You have taught me more than you know about patience with students and mentees, and it is a legacy I hope to continue.

Dr. Michael Medvinksy, for working with me extensively in the early stages of my research, getting me up to speed with the topics at hand, and for being a sounding board when new ideas came to light.

Dr. Eli Turkel, for giving his time to guide me and help my research along. You always brought new ideas to the table when I was not sure what approach to try next, and have been a valued member of our collaborations.

The rest of my committee, each of whom has influenced my academic career in their own ways. I had the privilege of studying under each of you through different courses during graduate school, and you have all graciously opened your office doors to me on numerous occasions to help with various academic and professional inquiries.

Everyone who has been there to support me and root for me through this whole journey: Eric Geiger, Zev Woodstock, Elizabeth Herman, and Katrina Petroske (to name a few) for all of the office shenanigans that helped break up the long hours in the office; My parents for all of the food and life lessons that got me this far; my church family at Lifepointe for the constant prayers, encouragement, and affirmations that have helped me grow so much in the last six years.

Last but certainly not least, my wife Rebecca. You have provided immeasurable support, encouragement, love, truth, and dog videos that have collectively gotten me through this journey. You make me a better man and scholar, and I cannot wait to spend the rest of my life with you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

1.1 Motivation

This dissertation introduces a non-iterative numerical method for solving non-overlapping domain decomposition problems. The application of focus for this work is the simulation of wave propagation in the frequency domain. One type of structure that is often used to simulate wave propagation is a waveguide. A waveguide is a structure that allows the propagation to occur only in a particular direction, such as sound passing through an instrument or air ducts. In the field of optics, a ring resonator (see Figure 1.1) is a closed loop waveguide paired with a light input and output. The ring's geometry determines certain resonant frequencies that propagate from the input, around the ring, and out at the output. Non-resonant frequencies will not propagate into and around the ring, so the ring resonator acts as a filter where a range of frequencies is the input into the system, only allowing the resonant frequencies to propagate to the output channel.

These resonators are used in a variety of optical applications, notably in fiber-optic cables. Some applications are in the field of biosensing [Chremmos et al., 2010; Fard et al., 2013], where ring resonators are integrated into micro-scale electronics. Such integrated waveguide ring resonators are known to suffer exponential propagation losses with the reduction of the ring radii. That sets a practical lower limit on the radius of the ring at a few micrometers [Chremmos et al., 2010]. For such micro-scale applications, photonic crystal ring resonators (PCRRs) offer a promising alternative, proving less sensitive to size-dependent losses [Chremmos et al., 2010]. PCRRs are constructed with periodically placed scattering rods where the exclusion of rods is used to form the path of the waveguide (Figure 1.2), with the rods themselves serving as a discrete alternative to a continuous wall. As a signal propagates through the waveguide, the rods refract and transmit the signal. This

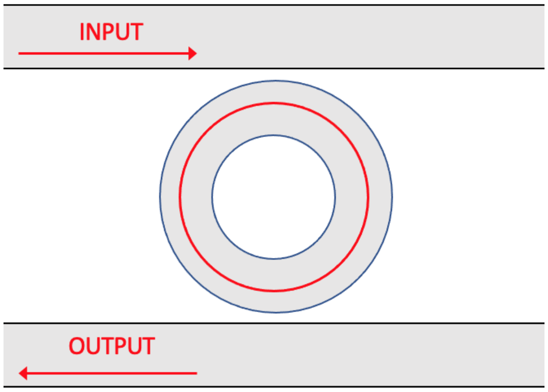


Figure 1.1 Example of a traditional ring resonator operating at a resonant frequency.

allows the signal to propagate down the open channels and into the central ring resonator in the usual fashion. Resonant frequencies can be tuned with the radius of the ring, as well as the size, material, and placement of the rods. Additionally, the periodic lattice of PCRRs can be structured with different symmetries (e.g., square, hexagonal, octagonal, etc...) [Romero-Vivas et al., 2005] depending on the application.

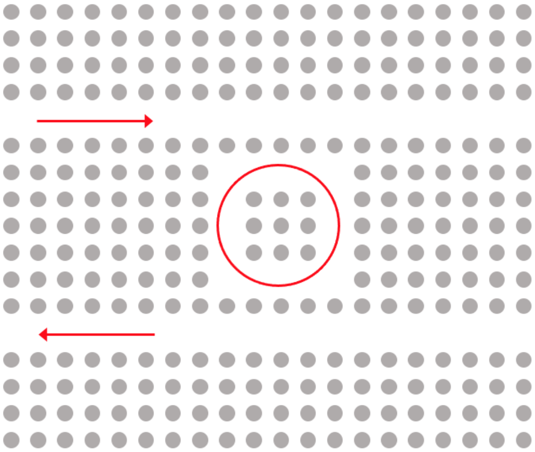


Figure 1.2 Layout of a photonic crystal ring resonator. Gray circles represent scattering rods and red indicates the path of a resonant signal.

Simulating the actual 3-dimensional structure is costly, so 2-dimensional simulations are commonly used as a surrogate for testing the efficacy of different arrangements of rods [Chremmos et al., 2010]. In the frequency domain, simulations involve solving the Helmholtz equation:

$$\Delta u + k^2 u = f, \quad x \in \Omega$$

where $k \in \mathbb{R}^+$ represents the wavenumber and is piecewise defined, representing the material properties of the background (typically air) or the rods individually. Due to the composite nature of the geometry of PCRRs, finite element methods have been the traditional solution method [Rodriguez-Esquerre et al., 2005; Rubio-Merccdes et al., 2004]. These involve sophisticated grids with high refinement levels to accurately resolve the space around each rod, leading to costly simulations.

Alternatively, the structure of the PCRR suggests the use of a domain decomposition method (DDM) where each rod is handled in its own subdomain (see Figure 1.3). Two known roadblocks for common iterative DDMs are cross-points (points where more than two subdomains meet) and discontinuous wavenumbers, both of which will be present in straightforward decompositions of a PCRR. Recent DDMs have found ways to mitigate these issues [Gander et al., 2016; D. Gordon et al., 2020], often through the use of context-dependent preconditioners or (computationally expensive) auxiliary variables. In our paper [North et al., 2022] (as well as Chapter 2 of this dissertation) we present a non-iterative DDM that is inherently insensitive to both cross-points and discontinuous wavenumbers. This insensitivity is due to the underlying use of the method of difference potentials (MDP) [Ryaben' kii, 2002].

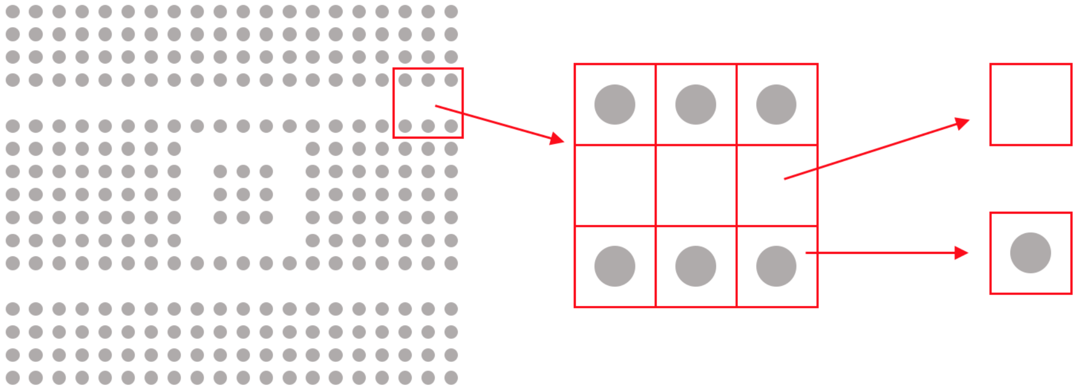
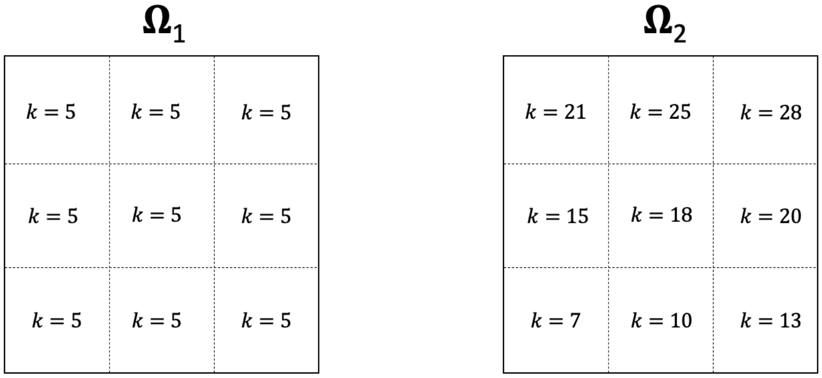


Figure 1.3 Depiction of how a PCRR can be broken into a small collection of basic subdomains (building blocks). Red represents artificial boundaries that are introduced to separate the subdomains from one another.

Originally proposed by Ryaben'kii [Ryaben' kii, 2002; Ryaben'kii, 1985], the MDP can be interpreted as a discrete version of the method of Calderon's operators [Calderón, 1963; Seeley, 1966] in the theory of partial differential equations. The MDP reduces a given partial differential equation from its domain to the boundary. The resulting boundary formulation involves an operator equation (Calderon's boundary equation with projection) with both Dirichlet and Neumann data in the capacity of unknowns. After solving the boundary operator equation, the solution is reconstructed on the domain using Calderon's potentials. Therefore, the MDP allows one to parameterize solutions on the domain using their boundary data. This proves convenient for a domain decomposition framework. Indeed, once the original domain has been partitioned into subdomains, the Calderon's boundary equations for individual subdomains are naturally coupled with the appropriate interface conditions that are also formulated in terms of the Dirichlet and Neumann data. This yields an overall linear system to be solved only at the combined boundary. The computation of the boundary projection operators is performed ahead of time and completely in parallel.

Rather than iteratively updating and matching boundary data along the interfaces as in traditional DDMs [Dolean et al., 2015], our method solves for all of the boundary and interface data simultaneously. The MDP benefits from the speed and convenience of a high-order finite difference method built on a regular, structured grid, even in the case of non-conforming geometries (such as the circular rods). Additionally, the MDP utilizes pre-computed operators that are built for a particular domain and unique for a given wavenumber. After the operators have been computed once they can simply be reused with new source or boundary data, assuming the geometry and wavenumber remain the same. In the context of DDMs, this property allows the same operators to be used for subproblems with the same shape and wavenumber. Our proposed decomposition of a PCRR only involves two kinds of subdomains — an empty square and a square with a rod, see Figure 1.3. By computing only two sets of operators, we obtain the pre-computed information for hundreds of subdomains for free.



Ω_1

$k = 5$	$k = 5$	$k = 5$
$k = 5$	$k = 5$	$k = 5$
$k = 5$	$k = 5$	$k = 5$

Ω_2

$k = 21$	$k = 25$	$k = 28$
$k = 15$	$k = 18$	$k = 20$
$k = 7$	$k = 10$	$k = 13$

Figure 1.4 The two extreme cases of domain decompositions in terms of how much time is saved when pre-computing the MDP operators. In Ω_1 all subdomains use the same operator so it is only computed once. In Ω_2 all subdomains have different wavenumbers, so the operators will need computed for all nine subdomains separately.

The pre-computation of operators is a major point of interest for decomposition methods. In traditional MDP settings, once an operator is computed for a particular domain and wavenumber, that operator can be reused for new problems that vary the source and boundary data. These operators are expensive to compute, so this can lead to substantial savings in computation time. Reusability of the operators extends to identical subdomains that share material properties (such as the wavenumber), allowing the computation of one operator in place of many. Of course, the actual savings in computation time will depend on how many qualitatively different subdomains are present in the decomposition. The bounding cases of the time savings are described in Figure 1.4, though most practical applications will fall between these cases.

The rest of this section covers the relevant background information for the numerical methods, excluding the MDP, utilized in the work of this dissertation. Chapter 2 presents the MDP in the context of a plain square and how to implement basic interface conditions between subdomains. Chapter 3 introduces the rod subdomain, which includes interface conditions between the rod and its surrounding subdomain as well as absorbing boundary conditions.

1.2 Background

1.2.1 Domain Decomposition

Domain Decomposition Methods were originally introduced by Schwarz [Schwarz, 1870] to prove the existence and uniqueness of solutions to the Poisson equation over irregularly shaped domains. The original Schwarz algorithm used overlapping decompositions (Figure 1.5), but was later extended to non-overlapping decompositions (Figure 1.6) by Lions [Lions, 1990]. In this work, we focus on the case of non-overlapping subdomains. Accordingly, this section begins by providing a brief overview of non-overlapping DDMs including the original method by Lions for the Poisson equation and subsequent adaptation by Després for the Helmholtz equation. For a more rigorous introduction to DDMs, including proofs of convergence and calculation of convergence factors, see [Dolean et al., 2015; Gander, 2008].

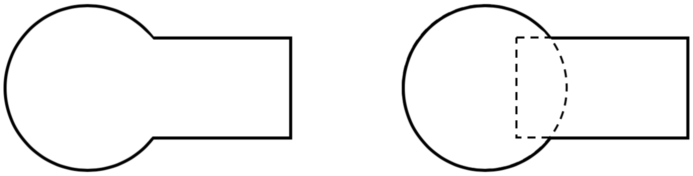


Figure 1.5 The classical DDM example. An irregular domain composed of two shapes, decomposed into two overlapping subdomains. Classical PDE analysis is non-trivial in the left domain, but is generally straightforward on shapes like circles and rectangles.

1.2.1.1 Non-Overlapping Formulation

Consider the Poisson equation over a rectangular domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$, (such as in Figure 1.6). Then the following Dirichlet boundary value problem (BVP) can be posed:

$$\begin{cases} \Delta u = f & \text{in } \Omega \\ u = 0 & \texttt{on } \partial\Omega \end{cases} \tag{1.1}$$

Consider a partitioning of Ω that splits the domain into two subdomains, Ω_1 and Ω_2, by introducing

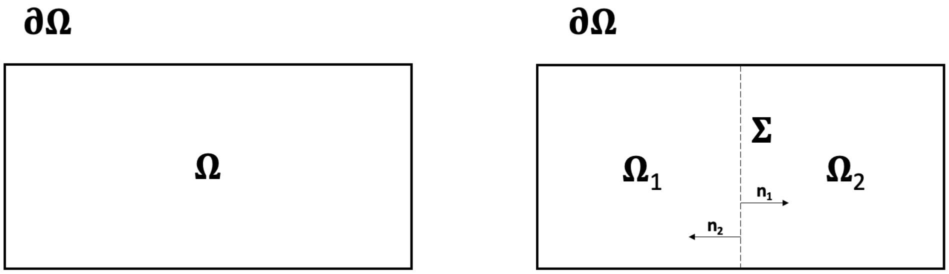


Figure 1.6 Basic non-overlapping decomposition of a domain Ω (with boundary $\partial\Omega$) into two subdomains, Ω_1 and Ω_2. A fictitious boundary, Σ, is introduced to indicate the separation between subdomains, and $\mathbf{n}_i$ is the outward unit normal vector of Ω_i on Σ.

an artificial interface $\Sigma = \overline{\Omega}_1 \cap \overline{\Omega}_2$ as in Figure 1.6. The BVP (1.1) can be reformulated over the new subdomains individually:

$$\begin{cases} \Delta u_1 = f & \text{in } \Omega_1 \\ u_1 = 0 & \text{on } \overline{\Omega}_1 \cap \partial\Omega \end{cases} \tag{1.2a}$$

$$\begin{cases} \Delta u_2 = f & \text{in } \Omega_2 \\ u_2 = 0 & \text{on } \overline{\Omega}_2 \cap \partial\Omega \end{cases} \tag{1.2b}$$

$$\begin{cases} u_1 = u_2, & \text{on } \Sigma \\ \frac{\partial u_1}{\partial \mathbf{n_1}} = -\frac{\partial u_2}{\partial \mathbf{n_2}} & \text{on } \Sigma \end{cases} \tag{1.2c}$$

where the interface conditions (1.2c) guarantee that the combined solution of (1.2) coincides with that of (1.1):

$$\begin{cases} u_1 = u, & \text{in } \Omega_1 \\ u_2 = u, & \text{in } \Omega_2 \end{cases}$$

Conditions other than (1.2c) can be formulated on Σ so that the resulting combined problem is well-posed, but its solution will be different from the true solution of (1.1).

There are two separate interface conditions in (1.2c). They apply to both subproblems (1.2a) and (1.2b) at the same time and couple these subproblems together. However, each subproblem (1.2a) or (1.2b) considered independently, i.e., with no connection to the other subproblem, is not fully specified and cannot be solved on its own because it is missing boundary conditions on Σ. To enable the individual solvability, one needs to provide these boundary conditions. Yet unlike in (1.2c), one cannot specify more than one boundary condition on Σ for either of the two standalone problems (1.2a) or (1.2b), as that would result in an overdetermination. In other words, when solving (1.2a) one cannot specify both u_1 and $\frac{\partial u_1}{\partial \mathbf{n_1}}$ along Σ, and likewise for (1.2b).

To avoid the overdetermination and still allow for separate solutions of individual subproblems, P.L. Lions proposed to use one Robin boundary condition [Lions, 1990], formed as a linear combination of the two continuity conditions (1.2c). For any pair of constants $\left(p_1, p_2\right) \in \mathbb{R}^2$, this *transmission condition* (or interface condition) yields the following combined formulation in lieu of (1.2):

$$\begin{cases} \Delta u_1 = f & \text{in } \Omega_1 \\ u_1 = 0 & \text{on } \overline{\Omega}_1 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_1}} + p_1\right) u_1 = \left(\frac{\partial}{\partial \mathbf{n_1}} + p_1\right) u_2 & \text{on } \Sigma \end{cases} \tag{1.3a}$$

$$\begin{cases} \Delta u_2 = f & \text{in } \Omega_2 \\ u_2 = 0 & \text{on } \overline{\Omega}_2 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_2}} + p_2\right) u_2 = \left(\frac{\partial}{\partial \mathbf{n_2}} + p_2\right) u_1 & \text{on } \Sigma \end{cases} \tag{1.3b}$$

Each of the two subproblems (1.3) is individually well-defined in the sense that the third equation in either (1.3a) or (1.3b) can be interpreted as a Robin boundary condition on Σ for u_1 or u_2, respectively, with the right-hand side of the respective equation providing the data. However, the relation of the combined formulation (1.3) to the original BVP (1.1) requires a special inquiry.

Lions conducted the corresponding analysis in [Lions, 1990]. He replaced the combined formulation (1.3) with the iteration:

$$\begin{cases} \Delta u_1^{(n+1)} = f & \text{in } \Omega_1 \\ u_1^{(n+1)} = 0 & \text{on } \overline{\Omega}_1 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_1}} + p_1\right) u_1^{(n+1)} = \left(\frac{\partial}{\partial \mathbf{n_1}} + p_1\right) u_2^{(n)} & \text{on } \Sigma \end{cases} \tag{1.4a}$$

$$\begin{cases} \Delta u_2^{(n+1)} = f & \text{in } \Omega_2 \\ u_2^{(n+1)} = 0 & \text{on } \overline{\Omega}_2 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_2}} + p_2\right) u_2^{(n+1)} = \left(\frac{\partial}{\partial \mathbf{n_2}} + p_2\right) u_1^{(n)} & \text{on } \Sigma \end{cases} \tag{1.4b}$$

and proved that this iteration converges to the solution of (1.1) as n increases. The rate of convergence depends on the choice of the parameters p_1 and p_2. As the next iteration $n+1$ for each subproblem only relies on the other subproblem's current iteration n, the subproblems can be solved in parallel to one another, a highly desirable trait for DDMs. The proof given in [Lions, 1990] extends to an arbitrary number of subdomains.

1.2.1.2 Helmholtz Adaptation

Complications arise when applying (1.4) directly to the Helmholtz equation. Consider the following BVP over the domain from Figure 1.6:

$$\begin{cases} \Delta u + k^2 u = f & \text{in } \Omega \\ u = 0 & \texttt{on } \partial\Omega \end{cases} \tag{1.5}$$

To guarantee well-posedness of (1.5), i.e., to avoid resonance, $-k^2$ may not be an eigenvalue of the underlying Laplace problem. However, when considering a decomposition such as the one in Figure 1.6 with the Lions transmission condition, it is non-trivial to know that $-k^2$ will always remain outside the spectrum of the corresponding Laplace subproblem, which only becomes more problematic when various decompositions are considered. This issue was addressed in [Després, 1993] when Després proposed the use of Lions' transmission condition with $p_1 = p_2 = ik$ (where $i = \sqrt{-1}$). This choice yields the following subproblems (cf. (1.3)):

$$\begin{cases} (\Delta + k^2)u_1 = f & \text{in } \Omega_1 \\ u_1 = 0 & \text{on } \overline{\Omega}_1 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_1}} + ik\right) u_1 = \left(\frac{\partial}{\partial \mathbf{n_1}} + ik\right) u_2 & \text{on } \Sigma \end{cases} \tag{1.6a}$$

$$\begin{cases} (\Delta + k^2)u_2 = f & \text{in } \Omega_2 \\ u_2 = 0 & \text{on } \overline{\Omega}_2 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_2}} + ik\right) u_2 = \left(\frac{\partial}{\partial \mathbf{n_2}} + ik\right) u_1 & \text{on } \Sigma \end{cases} \tag{1.6b}$$

Després' transmission condition shifts the spectrum of the operator to the complex domain, guaranteeing that resonant frequencies are avoided on each subproblem (1.6a) or (1.6b). It does so at the cost of introducing complex values into the problem, but for many applications, this is computationally not an issue. An iterative procedure similar to (1.4) can be employed for (1.6), and Després showed in [Després, 1993] that said procedure will converge.

1.2.1.3 Other Considerations

The methods outlined above are the foundation of most modern DDMs for the Helmholtz equation, and have been improved upon in recent years. For example, quasi-optimal convergence rates have been achieved by optimizing the choice of transmission conditions with the so-called "square root operator" [Boubendir et al., 2012]. However, while this leads to convergence in fewer iterations, it generally requires more expensive iterations.

Recent work has also been dedicated to the resolution of interior cross-points. The interior cross-points are points where more than two subdomains meet, and they pose no issues at the continuous level of the formulation. Yet the cross-points are known to adversely affect the accuracy and convergence if not discretized with care. In [Gander et al., 2016], several methods are discussed for resolving these cross-points for elliptic problems, and [Modave et al., 2020] provides an extension of the quasi-optimal method from [Boubendir et al., 2012] that accounts for interior cross-points. In Chapter 2, we demonstrate how the issue of cross-points is resolved naturally with our method, with no special consideration. While some other methods can also address the cross-points (see, for example, [D. Gordon et al., 2020]), we emphasize that our method is completely insensitive to them by design.

1.2.2 Absorbing Boundary Conditions

When simulating a PCRR, waves continue propagating toward infinity when they leave the domain of interest, contributing no reflections back into the domain of interest. In two dimensions, this effect is traditionally modeled by the Sommerfeld radiation condition [Sommerfeld, 1949]

$$\lim_{r\to\infty} \sqrt{r}\left(\frac{\partial}{\partial r} - ik\right) u(x,y) = 0, \quad r = \sqrt{x^2+y^2}. \tag{1.7}$$

This exact boundary condition is imposed at infinity, making it ill-suited for use in computational methods. Instead, we introduce an approximation of (1.7) called an *absorbing boundary condition* (ABC) at the edge of the domain. The simplest example of such an ABC is obtained by enforcing (1.7) on the boundary of the domain (rather than at infinity):

$$\left(\frac{\partial}{\partial \mathbf{n}} - ik\right) u = 0, \quad \text{on } \partial\Omega$$

where **n** represents the outward normal direction with respect to $\partial\Omega$. By its nature as an approximation, an ABC is inexact and will produce artificial reflections at the boundary which travel back into the domain of interest. A better quality ABC generates fewer reflections.

Various approaches have been taken to produce ABCs that reduce the artificial reflections, see approaches by Engquist and Majda [Engquist et al., 1977], Hagstrom and Hariharan [Hagstrom et al., 1988], and Bayliss et al. [Bayliss et al., 1982; Bayliss et al., 1980]. Additionally, see [S. V. Tsynkov, 1998] for a cohesive review of these and other methods.

We consider the second-order ABC developed by Engquist and Majda [Engquist et al., 1977]:

$$\mathscr{B}u \overset{\text{def}}{=} ik\frac{\partial u}{\partial \mathbf{n}} - k^2 u - \frac{1}{2}\frac{\partial^2 u}{\partial \tau^2} = 0 \tag{1.8}$$

where $\frac{\partial}{\partial \tau}$ represents differentiation with respect to the tangential coordinate. The normal and tangential coordinates used in (1.8) conveniently translate into Cartesian coordinates when the condition is set on the edge of a square. Details of how we implement (1.8) within our method can be found in Chapter 3.

1.2.3 Finite Difference Scheme

The MDP can be implemented in conjunction with any finite difference scheme as the underlying approximation, even in the case of complex or non-conforming boundaries. High-order schemes are effective in combating the well-known pollution effect [I. M. Babuška et al., 2000; Bayliss et al., 1983; Deraemaeker et al., 1999], generally achieving higher accuracy with coarser discretizations. Compact schemes do not require additional boundary conditions beyond those supplied with the differential equation itself. Therefore, we have chosen to use the fourth-order, compact scheme for the Helmholtz equation as presented in [Harari et al., 1995; Singer et al., 1998]:

$$\begin{aligned}
&\frac{1}{h^2}\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}-4u_{m,n}\right)\\
&+\frac{1}{6h^2}\big[u_{m+1,n+1}+u_{m-1,n+1}+u_{m+1,n-1}+u_{m-1,n-1}+4u_{m,n}\\
&-2\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}\right)\big]\\
&+\frac{k^2}{12}\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}+8u_{m,n}\right)\\
&=f_{m,n}+\frac{1}{12}\left(f_{m+1,n}+f_{m-1,n}+f_{m,n+1}+f_{m,n-1}-4f_{m,n}\right)
\end{aligned} \tag{1.9}$$

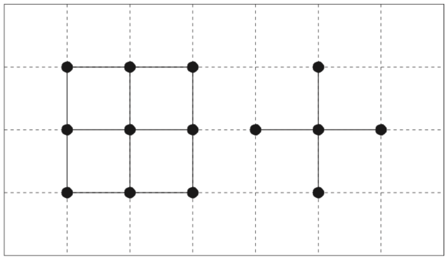


Figure 1.7 The stencils for the compact scheme given in (1.9).

The scheme (1.9) uses a 9-point stencil for the left-hand side of the PDE and a 5-point stencil for the right-hand side (see Figure 1.7) with uniform step size in the $x-$ and $y-$ directions ($\Delta x = \Delta y = h$). Additionally, (1.9) was derived with the assumption of a constant wavenumber, k. This assumption suffices for us because we assume that the decomposition yields subdomains that each have their own constant value of k. One could also consider a sixth-order scheme for constant [Turkel et al., 2013] or variable [Singer et al., 2006] wavenumber, or a fourth-order scheme for a more general form of the Helmholtz equation with a variable coefficient Laplace-like term and wavenumber [S. Britt et al., 2011]. However, for the scope of this dissertation we will focus on the case of piecewise constant k.

CHAPTER

2

UNIFORM SUBDOMAINS FOR DDM FRAMEWORK

The first building block to be introduced is the case of the empty subdomain - a square that contains no obstructing objects (such as a scattering rod). In the context of the Helmholtz equation these subdomains have a constant wavenumber, though they may have different wavenumbers from subdomain to subdomain. In this chapter, the formulation of the Method of Difference Potentials (MDP) is described for the case of a square domain. Details are then provided about how to combine more than one square in a domain decomposition formulation. This chapter is adapted from [North et al., 2022].

2.1 Method of Difference Potentials

To introduce the MDP [Ryaben' kii, 2002], consider the inhomogeneous Helmholtz equation with a general (constant-coefficient) Robin boundary condition

$$\Delta u + k^2 u = f \tag{2.1a}$$

$$\alpha u + \beta \frac{\partial u}{\partial \mathbf{n}} = \phi \tag{2.1b}$$

over the domain $\Omega \subset \mathbb{R}^2$ and its decomposition depicted in Figure 1.6. In a similar manner to traditional DDMs, we split the problem into two separate subdomains as in (1.2), and encounter the same issue of needing to enforce continuity of the solution and its flux over the interface Σ.

The key role of the MDP is to impose the required interface conditions on Σ. The MDP replaces the governing differential equation, the Helmholtz equation (2.1a), on the domain by an equivalent operator equation at the boundary (Calderon's boundary equation with projection). The latter is formulated with respect to the Cauchy data of the solution, i.e., the boundary trace of the solution itself (Dirichlet data) and its normal derivative (Neumann data). The reduction to the boundary is done independently for the individual subdomains Ω_1 and Ω_2 (see Figure 1.6). The resulting boundary equations with projections on the neighboring subdomains share the Dirichlet and Neumann data as unknowns at the common interface Σ, which directly enforces the continuity of the solution and its flux. For the remaining parts of the boundaries, $\partial\Omega_1\backslash\Sigma$ and $\partial\Omega_2\backslash\Sigma$, the boundary equations with projections are combined with the boundary condition (2.1b), which is formulated in terms of the Cauchy data of the solution. Altogether, the MDP solves a fully coupled problem for the Helmholtz equation similar to (1.2), and the constituent subproblems can still be solved independently and in parallel.

Indeed, Calderon's operators are computed with the help of the auxiliary problem, which is formulated for the same governing equation, but on a larger auxiliary domain. The auxiliary problem must be uniquely solvable and well-posed. Otherwise, the auxiliary problem can be arbitrary, and is normally chosen so as to enable an easy and efficient numerical solution. In particular, the auxiliary domain would typically have a simple, regular shape; some examples are shown in Figure 2.1. Given that for domain decomposition one needs to compute the Calderon operators separately for individual subdomains, we embed each subdomain within its own auxiliary domain, see Figure 2.2, and solve the resulting auxiliary problems independently. In practice, we also take advantage of the fact that in some of our simulations the subdomains are identical and reuse the computed operators accordingly. Precise criteria for the selection of an auxiliary domain, as well as details of how to efficiently account for identical subdomains, are discussed in Section 2.1.1.1.

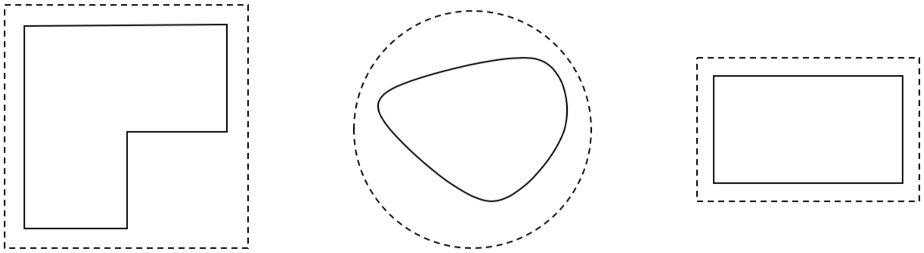


Figure 2.1 Example domains (solid border) and a reasonable choice of auxiliary domain (dotted border) for each.

In the rest of this section, those parts of the MDP necessary to implement it in the framework of DDM are introduced. For a detailed account of the theory and derivation of the MDP, see [Ryaben' kii, 2002], as well as [S. Britt et al., 2011; Medvinsky et al., 2012], among others. Additionally, for details

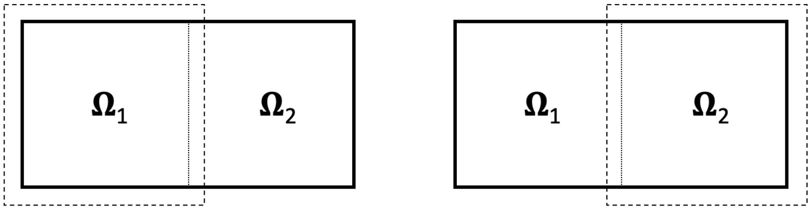


Figure 2.2 The auxiliary domain setup for our problem with the domain decomposition from Figure 1.6

on handling more complicated boundary conditions, as well as extending this method to domains with curvilinear sides, see [D. S. Britt et al., 2013] or [Medvinsky et al., 2016], respectively.

2.1.1 Base Subdomain

In this section, all of the components needed to perform the MDP algorithm on one base subdomain are defined. Considering the model domain from Figure 1.6, a logical choice of base subdomain is a square. For the sake of introducing the MDP on the base subdomain, throughout Section 2.1.1 we will refer to the base subdomain simply as Ω with boundary Γ, where Ω is a square with a side length of 2, centered about the origin. This simple case is used for efficiency. There is no substantial difficulty to treat alternative — including non-rectangular — subdivisions.

2.1.1.1 Auxiliary Problem

We will embed the base subdomain Ω in a larger domain Ω_0. This larger domain is known as the auxiliary domain, on which we formulate the auxiliary problem (AP). The AP should be uniquely solvable and well-posed, and should allow for a convenient and efficient numerical solution.

Let $\mathscr{L}$ represent the Helmholtz operator: $\mathscr{L}u \overset{\text{def}}{=} (\Delta + k^2 I)u$, where I is the identity operator. We formulate the AP on Ω_0 by supplementing the inhomogeneous Helmholtz equation with homogeneous Dirichlet conditions on the $y-$boundaries and local Sommerfeld-type conditions on the $x-$boundaries:

$$\begin{cases} \mathscr{L}u = g, & (x,y) \in \Omega_0, \\ u = 0, & y = \pm 1.1, \\ \frac{\partial u}{\partial x} + iku = 0, & x = 1.1, \\ \frac{\partial u}{\partial x} - iku = 0, & x = -1.1. \end{cases} \tag{2.2}$$

The choice of Sommerfeld-type conditions on the $x-$boundaries makes the spectrum of the AP (2.2) complex, guaranteeing that resonance is avoided for any real wavenumbers k. Hence, the AP (2.2) has a unique solution u for any right-hand side g. It should be noted that although similar in form to the Després condition from Section 1.2.1.2, the Sommerfeld-type conditions in (2.2) do not serve any transmission-related purpose, as they exist solely on the auxiliary domain and not on the physical boundary Γ.

To discretize the AP (2.2), we first replace the operator $\mathscr{L}$ with the left-hand side of the scheme (1.9). This creates a collection of equations centered at the node indexed by (m,n):

$$\begin{aligned}
&\frac{1}{h^2}\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}-4u_{m,n}\right)\\
&+\frac{1}{6h^2}\big[u_{m+1,n+1}+u_{m-1,n+1}+u_{m+1,n-1}+u_{m-1,n-1}+4u_{m,n}\\
&-2\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}\right)\big]\\
&+\frac{k^2}{12}\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}+8u_{m,n}\right)=g_{m,n}.
\end{aligned} \tag{2.3a}$$

To maintain the overall accuracy of the solution, the boundary conditions also need to be approximated to fourth-order. For the y—boundaries this is trivial, as the boundary nodes can directly be set to zero, i.e. for $m=0$, ..., M set

$$u_{m,0}=u_{m,N}=0. \tag{2.3b}$$

The following discretization of the Sommerfeld-type conditions was derived for the variable coefficient Helmholtz equation in [S. Britt et al., 2011] and simplified for the constant coefficient case in [D. S. Britt et al., 2013]:

$$\begin{aligned}
&\Bigg(\frac{u_{M,n}-u_{M-1,n}}{h}-\frac{1}{6h}\big(u_{M,n+1}-u_{M-1,n+1}+u_{M,n-1}-u_{M-1,n-1}\\
&-2\left(u_{M,n}-u_{M-1,n}\right)\big)-\frac{k^2h}{24}\left(u_{M,n}-u_{M-1,n}\right)\Bigg)\\
&+ik\Bigg(\frac{u_{M,n}-u_{M-1,n}}{h}+\frac{h^2k^2}{8}u_{M-\frac{1}{2},n}\\
&+\frac{u_{M-\frac{1}{2},n+1}-2u_{M-\frac{1}{2},n}+u_{M-\frac{1}{2},n-1}}{2}\Bigg)=0
\end{aligned} \tag{2.3c}$$

$$\begin{aligned}
&\Bigg(\frac{u_{1,n}-u_{0,n}}{h}-\frac{1}{6h}\left(u_{1,n+1}-u_{0,n+1}+u_{1,n-1}-u_{0,n-1}-2\left(u_{1,n}-u_{0,n}\right)\right)\\
&-\frac{k^2h}{24}\left(u_{1,n}-u_{0,n}\right)\Bigg)\\
&-ik\Bigg(\frac{u_{1,n}-u_{0,n}}{h}+\frac{h^2k^2}{8}u_{\frac{1}{2},n}+\frac{u_{\frac{1}{2},n+1}-2u_{\frac{1}{2},n}+u_{\frac{1}{2},n-1}}{2}\Bigg)=0.
\end{aligned} \tag{2.3d}$$

Conditions (2.3c) and (2.3d) were derived under the assumption that the source function is compactly supported. In our current setting, the grid function $g_{m,n}$ will be specified on the interior grid nodes, $m\in\{1,...,M-1\}$ and $n\in\{1,...,N-1\}$, and will be zero on the outermost grid nodes, $m\in\{0,M\}$ and $n\in\{0,N\}$.

We define the discrete operator $\mathscr{L}^{(h)}$ as the application of the left-hand side of (2.3a), allowing the discrete AP to be expressed as $\mathscr{L}^{(h)}u=g$ subject to the boundary conditions from (2.3b), (2.3c), and (2.3d). Similar to the continuous AP (2.2), the finite difference AP (2.3) has a unique solution u for any discrete right-hand side g. This solution u defines the inverse operator $\mathscr{G}^{(h)}$: $u=\mathscr{G}^{(h)}g$.

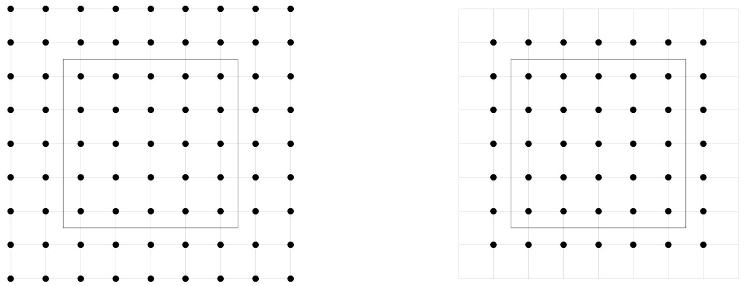
In particular, the right-hand side g may be defined as

$$g_{m,n} = \mathscr{B}^{(h)} f_{m,n} \overset{\text{def}}{=} f_{m,n} + \frac{1}{12}\big(f_{m+1,n} + f_{m-1,n} + f_{m,n+1} + f_{m,n-1} - 4f_{m,n}\big) \tag{2.4}$$

where $\mathscr{B}^{(h)}$ represents the application of the stencil from the right-hand side of the scheme (1.9). We emphasize that $\mathscr{G}^{(h)}$ is defined for any grid function g, not just those of the form $g = \mathscr{B}^{(h)} f$. The discrete AP can be solved by a combination of a sine-FFT in the $y-$direction and a tridiagonal solver in the $x-$direction to create an efficient approximation method for the solution to the continuous AP (2.2).

2.1.1.2 Grid Sets and Difference Potentials

Let $\mathbb{N}_0$ be a Cartesian grid on Ω_0 with uniform step size h in both the $x-$ and $y-$ directions. Let $\mathbb{M}_0 \subset \mathbb{N}_0$ be the set of nodes strictly interior to Ω_0, i.e. not on the boundary (see Figure 2.3). Define $\mathbb{M}^+ = \mathbb{M}_0 \cap \Omega$ as the nodes that are interior to the original domain Ω, and the exterior nodes as $\mathbb{M}^- = \mathbb{M}_0 \backslash \mathbb{M}^+$. Let $\mathbb{N}^+$ be the set of nodes needed to apply the 3×3 stencil from Figure 1.7 to every node in $\mathbb{M}^+$, and similarly let $\mathbb{N}^-$ be the same for $\mathbb{M}^-$ (see Figures 2.4a and 2.4b). Finally, we define the grid boundary $\gamma = \mathbb{N}^+ \cap \mathbb{N}^-$ as the discrete analogue of the original problem's boundary, Γ (see Figure 2.4c). Consider a grid function ξ_γ defined on the discrete boundary γ. We can then define the *difference potential with density* ξ_γ as



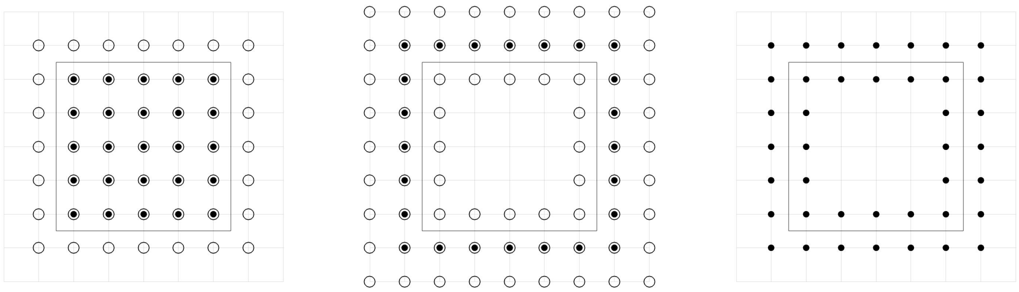
(a) $\mathbb{N}_0$

(b) $\mathbb{M}_0$

Figure 2.3 Cartesian grid sets used for the stencils presented in Figure 1.7 overlaid with the domain Ω.

$$P_{\mathbb{N}^+} \xi_\gamma \overset{\text{def}}{=} w - \mathscr{G}^{(h)}\big(\mathscr{L}^{(h)} w\big|_{\mathbb{M}^+}\big), \quad w = \begin{cases} \xi_\gamma & \text{on } \gamma, \\ 0 & \text{on } \mathbb{N}_0 \backslash \gamma. \end{cases} \tag{2.5}$$

The operation $\mathscr{L}^{(h)} w\big|_{\mathbb{M}^+}$ in (2.5) represents first applying the operator $\mathscr{L}^{(h)}$ to the grid function w, then truncating the result to the grid set $\mathbb{M}^+$. The difference potential $P_{\mathbb{N}^+} \xi_\gamma$ is a grid function defined on $\mathbb{N}^+$ (hence the notation). It satisfies the homogeneous finite difference equation $\mathscr{L}^{(h)}(P_{\mathbb{N}^+} \xi_\gamma) = 0$



(a) $\bullet - \mathbb{M}^+$, $\circ - \mathbb{N}^+$ **(b)** $\bullet - \mathbb{M}^-$, $\circ - \mathbb{N}^-$ **(c)** $\bullet - \gamma$

Figure 2.4 Discrete analogue grid sets of the interior, exterior, and boundary with respect to the original problem domain Ω.

on $\mathbb{M}^+$. By truncating the difference potential to the grid boundary, we obtain the projection operator P_γ:

$$P_\gamma \xi_\gamma \stackrel{\text{def}}{=} (P_{\mathbb{N}^+} \xi_\gamma)\big|_\gamma. \tag{2.6}$$

The projection P_γ defined by (2.6) has the following property: a grid function ξ_γ satisfies the difference Boundary Equation with Projection (BEP)

$$P_\gamma \xi_\gamma + Tr^{(h)} \mathscr{G}^{(h)} g = \xi_\gamma \tag{2.7}$$

if and only if there is a solution u on $\mathbb{N}^+$ to the finite difference equation (2.3a) such that ξ_γ is the trace of u on the grid boundary γ. In this case, u is reconstructed by means of the discrete generalized Green's formula

$$u = P_{\mathbb{N}^+} \xi_\gamma + \mathscr{G}^{(h)} g. \tag{2.8}$$

In particular, the discrete right-hand side g in equations (2.7) and (2.8) may be given by (2.4): $g = \mathscr{B}^{(h)} f$. Then, the discrete BEP (2.7) equivalently reduces the fourth order accurate discrete approximation of the Helmholtz equation $\mathscr{L}^{(h)} u = \mathscr{B}^{(h)} f$ from the grid domain $\mathbb{N}^+$ to the grid boundary γ. It will be convenient to specifically study the case where the governing equation is homogeneous, i.e. $f \equiv 0$. In this case, (2.7) reduces to

$$P_\gamma \xi_\gamma = \xi_\gamma. \tag{2.9}$$

Similar to (2.7) and (2.8), solutions of (2.9) can be used to reconstruct the corresponding solution u in the homogeneous case with the use of the difference potential

$$u = P_{\mathbb{N}^+} \xi_\gamma. \tag{2.10}$$

2.1.1.3 Equation-Based Extension

In order for u from (2.8) to approximate the solution of (2.1a) on $\mathbb{N}^+$, the grid density ξ_γ must be related, in a certain way, to the trace of the solution u at the continuous boundary Γ. This relation is expressed by the *extension operator*. Consider a pair of functions defined on Γ: $\xi_\Gamma = (\xi_0, \xi_1)\big|_\Gamma$. One can consider ξ_0 and ξ_1 as the Dirichlet and Neumann data, respectively, of some function $v = v(x, y)$ on Ω_0:

$$(\xi_0, \xi_1)\big|_\Gamma = \left(v, \frac{\partial v}{\partial \mathbf{n}}\right)\bigg|_\Gamma .$$

This function v can be defined in the vicinity of Γ as a truncated Taylor expansion, with ρ representing the distance (with sign) from the point of evaluation to Γ:

$$v(x,y) \overset{\text{def}}{=} v|_\Gamma + \rho \left.\frac{\partial v}{\partial \mathbf{n}}\right|_\Gamma + \frac{\rho^2}{2}\left.\frac{\partial^2 v}{\partial \mathbf{n}^2}\right|_\Gamma + \frac{\rho^3}{6}\left.\frac{\partial^3 v}{\partial \mathbf{n}^3}\right|_\Gamma + \frac{\rho^4}{24}\left.\frac{\partial^4 v}{\partial \mathbf{n}^4}\right|_\Gamma . \tag{2.11}$$

The definition (2.11) of the new function $v(x, y)$ is not complete until the higher order normal derivatives are provided. These can be obtained using equation-based differentiation applied to the Helmholtz equation (2.1a), where we assume v is a solution and v and $\frac{\partial v}{\partial \mathbf{n}}$ are known analytically on Γ. When the domain Ω is a square, the outward normal derivatives on Γ can be interpreted as standard $x-$ or $y-$ derivatives (or their negative counterparts), depending on which portion of the boundary one is considering.

For example, let the right side of the square be $x = X = \text{const}$. The outward normal derivative then coincides with the positive $x-$ derivative. By rearranging (2.1a) we immediately get an expression for the second $x-$derivative evaluated along Γ:

$$\frac{\partial^2 v}{\partial x^2}(X,y) = f(X,y) - \frac{\partial^2 v}{\partial y^2}(X,y) - k^2 v(X,y). \tag{2.12}$$

In this arrangement, $v(X, y)$ can be replaced with the known $\xi_0(y)$ and $\frac{\partial^2 v}{\partial y^2}(X, y)$ can be replaced with its second tangential derivative, $\xi_0''(y)$. The third and fourth derivatives can also be obtained by first differentiating (2.1a) with respect to x, then subsequently replacing $v(X, y)$ with $\xi_0(y)$, $\frac{\partial v}{\partial x}(X, y)$ with $\xi_1(y)$, and $\frac{\partial^2 v}{\partial x^2}(X, y)$ with the right-hand side of (2.12). This process yields the following expressions:

$$v(X,y) = \xi_0(y), \tag{2.13a}$$

$$\frac{\partial v}{\partial x}(X,y) = \xi_1(y), \tag{2.13b}$$

$$\frac{\partial^2 v}{\partial x^2}(X,y) = f(X,y) - \xi_0''(y) - k^2 \xi_0(y), \tag{2.13c}$$

$$\frac{\partial^3 v}{\partial x^3}(X,y) = \frac{\partial f}{\partial x}(X,y) - \xi_1''(y) - k^2 \xi_1(y), \tag{2.13d}$$

$$\frac{\partial^4 v}{\partial x^4}(X,y) = \frac{\partial^2 f}{\partial x^2}(X,y) - \frac{\partial^2 f}{\partial y^2}(X,y) - k^2 f(X,y) + \xi_0^{(4)}(y) + 2k^2 \xi_0^{(2)}(y) + k^4 \xi_0(y). \tag{2.13e}$$

The expressions in (2.13) can be substituted into (2.11) to calculate the values of $v(x, y)$ near the right side of Γ. Similar derivations can be used to compute $v(x, y)$ near other sides of the square, keeping in mind that the outward normal derivative on the left and bottom sides of the square correspond to the negative $x-$ and $y-$ derivatives, respectively.

The function $v = v(x, y)$ can be constructed starting from any pair of functions (ξ_0, ξ_1) defined on Γ by means of substituting (2.13a)-(2.13e) into the Taylor expansion (2.11). Then, sampling v only on the grid boundary γ, we define the extension operator **Ex** that yields the grid function ξ_γ as the following:

$$\xi_\gamma = \mathbf{Ex}(\xi_0, \xi_1) = v\Big|_\gamma.$$

As seen in (2.13), the operator **Ex** depends on the source term f. Hence, **Ex** is an affine operator:

$$\mathbf{Ex}\xi_\Gamma = \mathbf{Ex}^{(H)}(\xi_0, \xi_1) + \mathbf{Ex}^{(I)} f \tag{2.14}$$

where $\mathbf{Ex}^{(H)}$ represents its homogeneous (i.e., linear) part that only depends on (ξ_0, ξ_1), and $\mathbf{Ex}^{(I)}$ is the inhomogeneous part that accounts for the source term from (2.1a).

Although the formulae for the normal derivatives (2.13) were derived using the Helmholtz equation, $\xi_\Gamma = (\xi_0, \xi_1)$ does not need to represent the Cauchy data of a solution u to (2.1a) in order to apply the operator **Ex**. However, if ξ_Γ does correspond to a solution u: $\xi_\Gamma = \left(u, \frac{\partial u}{\partial \mathbf{n}}\right)\Big|_\Gamma$, then $\xi_\gamma = \mathbf{Ex}\,\xi_\Gamma$ approximates this solution near Γ with fifth-order accuracy with respect to the grid size h, specifically at the grid nodes of γ.

Let u be a solution to (2.1a) on Ω in the homogeneous case, $f \equiv 0$, and let ξ_Γ be the trace of u along the continuous boundary Γ such that $\xi_\Gamma = \left(u, \frac{\partial u}{\partial \mathbf{n}}\right)\Big|_\Gamma$. Let $\xi_\gamma = \mathbf{Ex}\xi_\Gamma$ and let $P_{\mathbb{N}^+}\xi_\gamma$ be the difference potential with density ξ_γ. Let p be the order of accuracy of the finite difference scheme (Sections 1.2.3 and 2.1.1.1). According to Reznik [Reznik, 1982, 1983] (alternatively, see [Ryaben' kii, 2002]), as the grid $\mathbb{N}_0$ is refined, $P_{\mathbb{N}^+}\xi_\gamma$ converges to the solution u (on the grid $\mathbb{N}^+$) with the convergence rate of $\mathcal{O}(h^p)$ provided that the number of terms in the Taylor expansion (2.11) is equal to $p + q$, where q is the order of the differential operator $\mathscr{L}$. Given that the Helmholtz equation is second-order and we use a fourth-order finite difference scheme (1.9), this would suggest the use of six terms in our extension. In practice, it has repeatedly been observed (see [Medvinsky et al., 2012], [Medvinsky et al., 2016], and [D. S. Britt et al., 2013], among others) that while sufficient, this bound is not tight, and the number of terms typically matches the order of accuracy of the finite difference scheme alone. Our use of four terms in (2.11) is corroborated by the numerical experiments in Section 2.2.

2.1.1.4 Series Representation of the Boundary Data

Consider a set of basis functions, $\{\psi_j\}$, and the following two sets of pairs

$$\psi_j^{(0)} = \left(\psi_j, 0\right), \quad \psi_j^{(1)} = \left(0, \psi_j\right), \quad j = 1, \ldots, \infty. \tag{2.15}$$

Recall that we denote the boundary data by $\xi_\Gamma = (\xi_0, \xi_1)$, where ξ_0 represents the Dirichlet data and ξ_1 represents the Neumann data. Specifically, consider one smooth section of Γ (i.e. one side of the square), denoted Γ*, and let its boundary data be denoted $\xi_{\Gamma^*} = \left(\xi_0^*, \xi_1^*\right)$. The separate components of this section of boundary data can be expanded individually along Γ*:

$$\xi_{\Gamma^*} = \left(\xi_0^*, \xi_1^*\right) = \sum_{j=1}^{\infty} \mathbf{c}_j^{(0)} \boldsymbol{\psi}_j^{(0)} + \sum_{j=1}^{\infty} \mathbf{c}_j^{(1)} \boldsymbol{\psi}_j^{(1)}. \tag{2.16}$$

The infinite series (2.16) can be truncated after a finite number of terms to provide an approximation of ξ_{Γ^*}. The number of terms M^* is typically taken so as to make the truncated terms negligible with respect to the accuracy attainable on the grid:

$$\xi_{\Gamma^*} = \sum_{j=1}^{M^*} \mathbf{c}_j^{(0)} \boldsymbol{\psi}_j^{(0)} + \sum_{j=1}^{M^*} \mathbf{c}_j^{(1)} \boldsymbol{\psi}_j^{(1)}. \tag{2.17}$$

Provided that the boundary data are sufficiently smooth, for the appropriately chosen basis functions ψ_j (e.g. Chebyshev, Fourier, etc...) the value of M^* can be taken relatively small. The series representation (2.17) can be extended to apply to all four sides of the square Γ by combining the corresponding basis functions. Consider the labeling of the sides in Figure 2.5, and the following definition of the expanded set of basis functions Ψ_j:

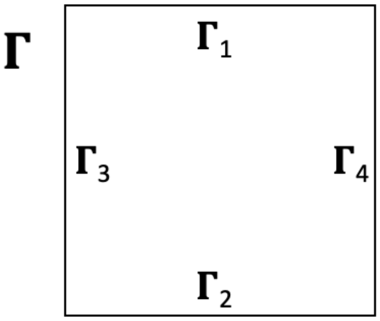


Figure 2.5 A labeling of the sides of Γ. The choice of ordering is arbitrary and only given here as a visual reference for the linear system in Section 2.1.1.5.

$$\Psi_{j+(i-1)M^*} = \begin{cases} \psi_j, & \text{on } \Gamma_i \\ 0, & \text{otherwise} \end{cases} \qquad \text{for } i = 1, \ldots, 4. \tag{2.18}$$

Every element of Ψ in (2.18) is defined on the entire Γ, while each $\xi_{\Gamma i}$ has a series expansion independent of the others. Then, similar to (2.15) we define the following pairs:

$$\boldsymbol{\Psi}_j^{(0)} = \left(\Psi_j, 0\right), \quad \boldsymbol{\Psi}_j^{(1)} = \left(0, \Psi_j\right), \quad j = 1, ..., M$$

where $M = 4 \cdot M^*$, and write the expansion of ξ_Γ as

$$\xi_\Gamma = \sum_{j=1}^{M} \mathbf{c}_j^{(0)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \boldsymbol{\Psi}_j^{(1)}. \tag{2.19}$$

Note that the choice of the same system of basis function for both the Dirichlet and Neumann data, as well as for all four sides of the square, is not a requirement. However, it provides extra convenience for constructing the linear system in Section 2.1.1.5 and building the DDM extension in Section 2.1.2.

2.1.1.5 Forming the Base Linear System

Applying the extension operator (2.14) to the series representation of ξ_Γ in (2.19), we have:

$$\begin{aligned}\mathbf{Ex}\,\xi_\Gamma &= \mathbf{Ex}^{(H)}\left(\sum_{j=1}^{M} \mathbf{c}_j^{(0)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \boldsymbol{\Psi}_j^{(1)}\right) + \mathbf{Ex}^{(I)} f \\ &= \sum_{j=1}^{M} \mathbf{c}_j^{(0)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(1)} + \mathbf{Ex}^{(I)} f. \end{aligned} \tag{2.20}$$

Setting $\xi_\gamma = \mathbf{Ex}\,\xi_\Gamma$ and substituting it into the BEP (2.7) with $g = \mathscr{B}^{(h)} f$ yields:

$$\begin{aligned}
&P_\gamma \xi_\gamma = \xi_\gamma - Tr^{(h)} \mathscr{G}^{(h)} \mathscr{B}^{(h)} f \\
&P_\gamma \mathbf{Ex}\,\xi_\Gamma = \mathbf{Ex}\,\xi_\Gamma - Tr^{(h)} \mathscr{G}^{(h)} \mathscr{B}^{(h)} f \\
&P_\gamma \left(\sum_{j=1}^{M} \mathbf{c}_j^{(0)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(1)} + \mathbf{Ex}^{(I)} f\right) \\
&\qquad = \sum_{j=1}^{M} \mathbf{c}_j^{(0)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(1)} + \mathbf{Ex}^{(I)} f - Tr^{(h)} \mathscr{G}^{(h)} \mathscr{B}^{(h)} f \\
&\sum_{j=1}^{M} \mathbf{c}_j^{(0)} P_\gamma \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} P_\gamma \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(1)} + P_\gamma \mathbf{Ex}^{(I)} f \\
&\qquad = \sum_{j=1}^{M} \mathbf{c}_j^{(0)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(1)} + \mathbf{Ex}^{(I)} f - Tr^{(h)} \mathscr{G}^{(h)} \mathscr{B}^{(h)} f.
\end{aligned}$$

By collecting similar terms, we obtain the following system of linear algebraic equations

$$\sum_{j=1}^{M}\mathbf{c}_j^{(0)}(P_\gamma - I_\gamma)\mathbf{Ex}^{(H)}\mathbf{\Psi}_j^{(0)} + \sum_{j=1}^{M}\mathbf{c}_j^{(1)}(P_\gamma - I_\gamma)\mathbf{Ex}^{(H)}\mathbf{\Psi}_j^{(1)}$$
$$= (I_\gamma - P_\gamma)\mathbf{Ex}^{(I)}f - Tr^{(h)}\mathscr{G}^{(h)}\mathscr{B}^{(h)}f \tag{2.21}$$

where I_γ represents the identity operator in the space of grid functions ξ_γ defined on γ. System (2.21) can be written in matrix form:

$$Qc = F \tag{2.22}$$

where the matrix $Q = \left[Q^{(0)}, Q^{(1)}\right]$ is given by

$$Q = \Big[\underbrace{(P_\gamma - I_\gamma)\mathbf{Ex}\mathbf{\Psi}_1^{(0)}, \dots, (P_\gamma - I_\gamma)\mathbf{Ex}\mathbf{\Psi}_M^{(0)},}_{Q^{(0)}}$$
$$\underbrace{(P_\gamma - I_\gamma)\mathbf{Ex}\mathbf{\Psi}_1^{(1)}, \dots, (P_\gamma - I_\gamma)\mathbf{Ex}\mathbf{\Psi}_M^{(1)}}_{Q^{(1)}}\Big]. \tag{2.23}$$

This matrix has dimension $|\gamma| \times 2M$ where $|\gamma|$ is the number of nodes in the grid boundary γ. The column vector

$$c = [\underbrace{c_1, \dots, c_M}_{c^{(0)\mathsf{T}}}, \underbrace{c_{M+1}, \dots, c_{2M}}_{c^{(1)\mathsf{T}}}]^{\mathsf{T}} \tag{2.24}$$

in equation (2.22) is a vector of unknowns with dimension $2M$, while the vector F has dimension $|\gamma|$ and represents the inhomogeneous part of the problem: $(I_\gamma - P_\gamma)\mathbf{Ex}^{(I)}f - Tr^{(h)}\mathscr{G}^{(h)}\mathscr{B}^{(h)}f$. The first M columns of Q in (2.23) form the sub-matrix $Q^{(0)}$ and correspond to the coefficients $\{c_j^{(0)}\}$ in (2.24), while columns $M+1$ through $2M$ form $Q^{(1)}$ and correspond to $\{c_j^{(1)}\}$.

Note that, solution to (2.22) is not unique, as the system is derived only from (2.7), (2.4) and does not take into account any boundary conditions. Therefore, we interpret the underdetermined system (2.22) as a core piece of the multi-subdomain decomposition algorithm, rather than a system to be solved in its own right. The decomposition algorithm is described in Section 2.1.2 for the case of two subdomains and subsequently extended in Section 2.1.3 to the case of a larger number of subdomains. For a discussion about implementing boundary conditions and completing the algorithm in the single domain case, see [D. S. Britt et al., 2013; S. Britt et al., 2011; Medvinsky et al., 2012, 2016; Ryaben' kii, 2002].

2.1.2 Extension to 2 Subdomains

Reconsider the problem of solving (2.1) over a partitioned domain as in Figure 1.6. Let $\Gamma^{(i)}$ represent the boundary of Ω_i, and let each $\Gamma^{(i)}$ be composed of its four sides as in Figure 2.5, so that $\Gamma^{(i,j)}$ denotes side j of $\Gamma^{(i)}$. Further, define a new set of indices, B, to be the indices of $\Gamma^{(i,j)}$ that correspond to the boundary edges. For the two-domain case, this yields $B = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,4)\}$, as well as its complement $B^{\complement} = \{(1,4),(2,3)\}$ for the indices corresponding to both sides of the interface Σ.

Let all grid sets and operators from Section 2.1.1 be defined for Ω_1 and Ω_2, independently. Partition and index the matrix Q and the unknown column vector c with the following notation:

$$\begin{aligned} Q^{(i,*,k)} &= \begin{bmatrix} Q^{(i,1,k)} & Q^{(i,2,k)} & Q^{(i,3,k)} & Q^{(i,4,k)} \end{bmatrix} \\ c^{(i,*,k)} &= \begin{bmatrix} c^{(i,1,k)\mathsf{T}} & c^{(i,2,k)\mathsf{T}} & c^{(i,3,k)\mathsf{T}} & c^{(i,4,k)\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \end{aligned}$$

For $Q^{(i,j,k)}$, the indices $i \in \{1,2\}$ and $j \in \{1,2,3,4\}$ denote those columns corresponding to the basis functions defined to be non-zero over $\Gamma^{(i,j)}$. The index $k \in \{0,1\}$ distinguishes between the Dirichlet and Neumann data (compare to the notation $Q^{(0)}$ and $Q^{(1)}$ in Section 2.1.1.5). The use of $c^{(i,j,k)}$ similarly identifies the coefficients of the corresponding basis functions over $\Gamma^{(i,j)}$ in either the Dirichlet or Neumann case. The independent linear systems for Ω_1 and Ω_2 can then be written as

$$Q^{(1,*,*)} c^{(1,*,*)} = F^{(1)} \quad \text{and} \quad Q^{(2,*,*)} c^{(2,*,*)} = F^{(2)}$$

where $Q^{(i,*,*)} = \begin{bmatrix} Q^{(i,*,0)} & Q^{(i,*,1)} \end{bmatrix}$ and $c^{(i,*,*)} = \begin{bmatrix} c^{(i,*,0)\mathsf{T}} & c^{(i,*,1)\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$. Note that the construction of $Q^{(i,*,*)}$ is identical to that of (2.23) over a single subdomain. Equivalently, the independent linear systems can be expressed simultaneously as the block-diagonal system

$$\begin{bmatrix} Q^{(1,*,*)} & 0 \\ 0 & Q^{(2,*,*)} \end{bmatrix} \begin{bmatrix} c^{(1,*,*)} \\ c^{(2,*,*)} \end{bmatrix} = \begin{bmatrix} F^{(1)} \\ F^{(2)} \end{bmatrix}. \tag{2.25}$$

Similar to (2.22), the solution to (2.25) is not unique because it is derived only from the discrete BEP (2.7) combined with (2.4) for each subdomain and does not account for the boundary condition (2.1b). Additionally, since Ω has been decomposed into Ω_1 and Ω_2, an interface (or transmission) condition is needed to account for the lack of a boundary condition along the interface Σ.

2.1.2.1 Boundary Conditions

To account for the boundary conditions, consider one $\Gamma^{(i,j)}$ (for $(i,j) \in B$). Substitute the series representation at the boundary (cf. (2.17)) into the boundary condition (2.1b) for both u and $\frac{\partial u}{\partial \mathbf{n}}$. Expand the right-hand side of (2.1b) as $\phi = \sum_m^{M^*} d_m^{(i,j)} \psi_m$ (using the same basis functions as in (2.15)). Then,

$$\alpha \left(\sum_{m=1}^{M^*} c_m^{(i,j,0)} \psi_m \right) + \beta \left(\sum_{m=1}^{M^*} c_m^{(i,j,1)} \psi_m \right) = \sum_m^{M^*} d_m^{(i,j)} \psi_m. \tag{2.26}$$

Assuming that the basis functions ψ_m are orthogonal, we derive from (2.26):

$$\alpha c_m^{(i,j,0)} + \beta c_m^{(i,j,1)} = d_m^{(i,j)}, \quad \text{for } m \in \{1, ..., M^*\}. \tag{2.27}$$

The M^* equations (2.27) can be obtained for each index pair in B, adding a total of $6M^*$ extra equations. Note here that the sets of equations obtained for each $\Gamma^{(i,j)}$ are independent from one another, allowing greater flexibility in the boundary condition (2.1b). For example, the definitions of α and β in (2.1b) can be piece-wise constant, split along the different sections of Γ, i.e. on $\Gamma^{(i,j)}$

$$\alpha = \alpha^{(i,j)} \quad \beta = \beta^{(i,j)} \tag{2.28}$$

where $\alpha^{(i,j)}$ and $\beta^{(i,j)}$ are constants and $\left(\alpha^{(i,j)}, \beta^{(i,j)}\right) \neq (0,0)$ for any pair $(i, j) \in B$. This generalization allows for both Dirichlet and Neumann conditions ($\beta^{(i,j)} = 0$ or $\alpha^{(i,j)} = 0$, respectively) as particular cases. The equations being added by (2.27) are sparse compared to the rest of (2.25), which can be taken advantage of computationally, see Section 2.1.2.3. Further information on implementing mixed boundary conditions, as well as extending this process to include variable coefficient Robin conditions, can be found in [D. S. Britt et al., 2013].

2.1.2.2 Interface Condition

The standard interface condition requires continuity of the solution and its flux across the interface (see Section 1.2.1.1). These two conditions can be enforced by equating the series representations of the Dirichlet data along $\Gamma^{(1,4)}$ and $\Gamma^{(2,3)}$, as well as setting the series representation for the Neumann data of $\Gamma^{(1,4)}$ equal to the negative of that for $\Gamma^{(2,3)}$:

$$\sum_{m=1}^{M^*} c_m^{(1,4,0)} \psi_m = \sum_{m=1}^{M^*} c_m^{(2,3,0)} \psi_m \tag{2.29a}$$

$$\sum_{m=1}^{M^*} c_m^{(1,4,1)} \psi_m = -\sum_{m=1}^{M^*} c_m^{(2,3,1)} \psi_m. \tag{2.29b}$$

As with the boundary conditions in Section 2.1.2.1, the use of identical sets of orthogonal basis functions along each side is exploited to obtain the following two sets of equations for each $m \in \{1, ..., M^*\}$:

$$c_m^{(1,4,0)} - c_m^{(2,3,0)} = 0, \tag{2.30a}$$

$$c_m^{(1,4,1)} + c_m^{(2,3,1)} = 0. \tag{2.30b}$$

Collectively, (2.30) provides $2M^*$ equations to supplement (2.25).

Alternative interface conditions can be chosen and implemented in a similar fashion. For example, if u_i is the solution to the subproblem on Ω_i, then for constants $a^{(0)}, a^{(1)}, b^{(0)}, b^{(1)}$ and smooth functions $\eta^{(0)}, \eta^{(1)}$, a class of interface conditions can be defined as follows on the interface $\Gamma^{(1,4)} = \Gamma^{(2,3)}$:

$$a^{(0)} u_1 + b^{(0)} u_2 = \eta^{(0)} \qquad a^{(1)} \frac{\partial u_1}{\partial \mathbf{n_1}} + b^{(1)} \frac{\partial u_2}{\partial \mathbf{n_2}} = \eta^{(1)}. \tag{2.31}$$

Any transmission conditions of type (2.31) can be accounted for by following the same steps as in (2.29) and (2.30). The only addition is to let $\eta^{(0)} = \sum_{m=1}^{M^*} \eta_m^{(0)} \psi_m$ and $\eta^{(1)} = \sum_{m=1}^{M^*} \eta_m^{(1)} \psi_m$ be the expansions of $\eta^{(0)}$ and $\eta^{(1)}$. This yields

$$\begin{aligned} a^{(0)} c_m^{(1,4,0)} + b^{(0)} c_m^{(2,3,0)} &= \eta_m^{(0)} \\ a^{(1)} c_m^{(1,4,1)} + b^{(1)} c_m^{(2,3,1)} &= \eta_m^{(1)} \end{aligned}$$

as the conditions for the coefficients, where the choice of $a^{(0)} = a^{(1)} = b^{(1)} = 1$, $b^{(0)} = -1$, and $\eta^{(0)} \equiv \eta^{(1)} \equiv 0$ recovers the original condition. By allowing linear combinations and inhomogeneities in the interface conditions, a wider set of situations such as jumps over the interface in the solution, its flux, or both can be accounted for. In this work, however, we are only considering the case where the solution and its flux are continuous on Σ.

2.1.2.3 Solving the Complete System

By supplementing the system (2.25) with the equations derived in (2.27) and (2.30), the complete system can be expressed as a new matrix equation

$$\overline{Q} c^{(*,*,*)} = \overline{F} \tag{2.32}$$

where the dimension of $\overline{Q}$ is $(2|\gamma| + 8M^*) \times 16M^*$. The system (2.32) can be solved by minimizing the ℓ_2 norm $\|\overline{Q} c^{(*,*,*)} - \overline{F}\|_2$ through traditional least squares methods, e.g., a QR-factorization. Note that while the least squares solution is unique, it is the existence of a classical solution to (2.1) — from which (2.32) is ultimately derived — that guarantees $\|\overline{Q} c^{(*,*,*)} - \overline{F}\|_2$ will be within discretization error of zero. In fact, if M^* is chosen large enough in (2.17), then $\|\overline{Q} c^{(*,*,*)} - \overline{F}\|_2$ decreases at a rate of $\mathcal{O}(h^4)$ (the order of accuracy of the finite difference scheme) as the grid is refined.

Rather than adding equations (2.27) and (2.30) to the system, these conditions can instead be resolved through substitution and the elimination of unknowns. For the boundary condition equations (2.27), first consider the case where (2.1b) reduces to a Dirichlet boundary condition (i.e. $\alpha = 1$, $\beta = 0$). In this case, the coefficients $c^{(i,j,0)}$ (for $(i,j) \in B$) are obtained directly when expanding the right-hand side of (2.1b), eliminating those coefficients from the larger linear system. The coefficients in $c^{(i,j,0)}$ are multiplied by the corresponding columns of $Q^{(i,j,0)}$, then subtracted over to the right-hand side of (2.25). If (2.1b) reduces to a Neumann condition (i.e. $\alpha = 0$, $\beta = 1$), the same process is followed but for $c^{(i,j,1)}$ and $Q^{(i,j,1)}$. In either case, this process eliminates $6M^*$ unknowns from the system (M^* unknowns for each $(i,j) \in B$) leaving $10M^*$ unknowns rather than the original $16M^*$ unknowns.

When (2.1b) does not reduce to a Dirichlet or Neumann condition ($\alpha \neq 0$ and $\beta \neq 0$), we can still eliminate unknowns by means of substitution. Consider (2.27), and rearrange the terms to solve for either $c_m^{(i,j,0)}$ or $c_m^{(i,j,1)}$:

$$c_m^{(i,j,1)} = \frac{1}{\beta} d_m^{(i,j)} - \frac{\alpha}{\beta} c_m^{(i,j,0)}. \tag{2.33}$$

From (2.33), the $\frac{1}{\beta} d_m^{(i,j)}$ terms can be multiplied by the corresponding columns of $Q^{(i,j,1)}$ and subtracted to the right-hand side, while the $\frac{\alpha}{\beta} c_m^{(i,j,0)}$ terms can be combined with their like terms from the original system (2.25). Similar to the Dirichlet and Neumann cases, $6M^*$ unknowns are eliminated from the system.

The interface conditions (2.30a) can be accounted for by adding the respective columns, $Q_m^{(1,4,0)}$ and $Q_m^{(2,3,0)}$, and eliminating one of the coefficients, $c_m^{(1,4,0)}$ or $c_m^{(2,3,0)}$. As these conditions exist for $m \in \{1, \ldots, M^*\}$, resolving the interface conditions this way eliminates M^* unknowns from the system. Following the same process for (2.30b) (subtracting columns instead of adding) eliminates an additional M^* unknowns.

In the case where (2.27) and (2.30) are included as supplemental equations, the overall system has dimension $(2|\gamma| + 8M^*) \times 16M^*$. If the conditions are resolved, the dimension is reduced to $2|\gamma| \times 8M^*$, which enables faster solution. The solution vector $c^{(*,*,*)}$ is used to reconstruct $\xi_{\Gamma^{(1)}}$ and $\xi_{\Gamma^{(2)}}$ through the series representation (2.17) for each subproblem. In turn, $\xi_{\Gamma^{(1)}}$ and $\xi_{\Gamma^{(2)}}$ are extended to their respective grid boundaries, as described in Section 2.1.1.3. Finally, a fourth-order accurate approximation to the unique solution of (2.1) is obtained by applying (2.8) to the resulting $\xi_{\gamma^{(1)}}$ and $\xi_{\gamma^{(2)}}$. These approximations collectively provide an approximation of the global solution to (2.1) on the overall domain Ω.

2.1.3 Extension to N Subdomains

The extension to N subdomains is a natural extension of the two-subdomain case. Consider (2.1) over a domain Ω that is split into N identical (square) subdomains, whose interfaces are full edges of the squares (see Figure 2.6). Returning to the triple index notation used in Section 2.1.2, let the first argument vary from 1 to N, rather than stopping at 2, and let all grid sets and operators from Section 2.1.1 be defined independently for each Ω_i. To build the matrix for the linear system, combine the $Q^{(i,*,*)}$ from each subdomain in a block-diagonal style. The vectors of unknowns and

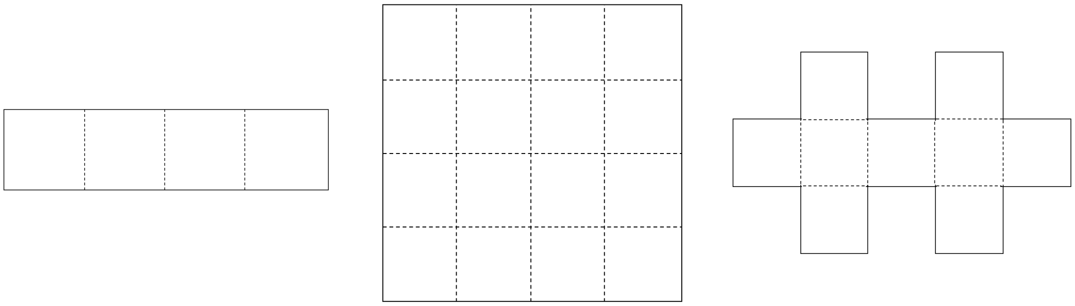


Figure 2.6 Examples of domains that have a valid $N-$subdomain decomposition.

right-hand sides from each subdomain are simply appended to create the following system:

$$\begin{bmatrix} Q^{(1,*,*)} & 0 & 0 & 0 \\ 0 & Q^{(2,*,*)} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Q^{(N,*,*)} \end{bmatrix} \begin{bmatrix} c^{(1,*,*)} \\ c^{(2,*,*)} \\ \vdots \\ c^{(N,*,*)} \end{bmatrix} = \begin{bmatrix} F^{(1)} \\ F^{(2)} \\ \vdots \\ F^{(N)} \end{bmatrix}. \tag{2.34}$$

To generalize the handling of boundary and interface conditions, extend the definition of the set B

$$B = \left\{(i,j) \middle| \Gamma^{(i,j)} \cap \partial\Omega \neq \emptyset \right\} \tag{2.35}$$

so that $|B \cup B^{\complement}| = 4N$. If $(i,j) \in B$, then $\Gamma^{(i,j)}$ has an associated boundary condition specified by (2.1b) and the process described in Section 2.1.2.1 can be applied for each $(i,j) \in B$ to obtain the necessary supplemental equations. If $(i,j) \in B^{\complement}$, then $\Gamma^{(i,j)}$ is an interface, requiring the process from Section 2.1.2.2 to determine the supplemental equations. Adding these equations yields the N subdomain version of (2.32):

$$\overline{Q}_N c^{(*,*,*)} = \overline{F}_N \tag{2.36}$$

where $\overline{Q}_N$ and $\overline{F}_N$ represent the matrix from the left-hand side of (2.34) and the vector of the right-hand side, respectively, after being supplemented with boundary and interface condition equations. The shape of the domain determines how many equations correspond to boundary conditions as opposed to interface conditions, but there will always be $4NM^*$ equations added to (2.34) (M^* equations for each $\Gamma^{(i,j)}$). As in Section 2.1.2.3, these equations can often be resolved with substitution and elimination to reduce the cost of solving the linear system.

2.1.4 Implementation Details

In this section, we provide the important implementation details of the proposed algorithm, which are further justified in Section 2.1.5. We assume that all subdomains are identical squares and that Ω satisfies the requirements described at the beginning of Section 2.1.3 and in Figure 2.6. In (2.1a), we assume that the wavenumber k is piece-wise constant over Ω, and constant on any given Ω_i. Such assumptions allow the exploration of the best-case scenario. A brief discussion of possible generalizations is given in Section 2.4. Consider the following summary of the algorithm:

1. For each Ω_i:
   (a) Define the auxiliary problem (2.2), as well as the grid sets and operators from Section 2.1.1.2.
   (b) Compute $Q^{(i,*,*)}$, the left-hand side of (2.22).
   (c) Compute $P_\gamma \mathbf{Ex}^{(I)} f$ and $Tr^{(h)} \mathscr{G}^{(h)} \mathscr{B}^{(h)} f$ to form $F^{(i)}$ from the right-hand side of (2.21).
2. Solve (2.36) for $c^{(*,*,*)}$:

(a) Assemble (2.34) and resolve the boundary conditions and interface conditions (either as supplemental equations or as in Section 2.1.2.3).

(b) Compute the QR-factorization of $\overline{Q}_N = QR$.

(c) Compute $c^{(*,*,*)} = R^{-1}Q^*\overline{F}_N$ where Q^* is the conjugate transpose of Q.

3. For each Ω_i:

   (a) Use $c^{(i,*,*)}$ and the series representation (2.17) for each $\Gamma^{(i,j)}$ to reconstruct the boundary data.

   (b) Extend $\xi_{\Gamma^{(i)}}$ using **Ex** to obtain ξ_γ.

   (c) Apply (2.8) to ξ_γ to obtain a local, fourth-order approximation to the solution u on Ω_i.

The local solutions are then assembled to collectively provide a global approximation of the solution u on Ω. Note, that the entirety of steps 1 and 3 can be distributed on parallel processors for each subdomain. Once the algorithm has been run, the structure of the method allows several problem variations to be solved more economically because they do not affect the structure of terms that have already been computed in specific parts of steps 1 and 2.

The first part of the algorithm that requires special consideration is step 1(b). Constructing $Q^{(i,*,*)}$ is expensive (see Section 2.1.5), but in general we do not need to recompute it every time the algorithm is run. Due to the use of geometrically identical subdomains as well the same set of basis functions throughout the problem, the only factor that distinguishes $Q^{(i,*,*)}$ from $Q^{(j,*,*)}$ is the wavenumber k on Ω_i and Ω_j. Therefore, $Q^{(i,*,*)}$ can be reused for any subdomain Ω_j such that the value of k is shared across both Ω_i and Ω_j. The cost to construct $Q^{(i,*,*)}$ should only be accrued once for each unique value of k across all subdomains. In the case where k is uniform across Ω, $Q^{(i,*,*)} = Q^{(j,*,*)}$ for all $i, j \in \{1,...,N\}$, so the base linear system is only computed once, regardless of the number of subdomains. Further, as long as each $Q^{(i,*,*)}$ is saved after being computed, it can be reused in future problems for subdomains with the corresponding value of k, thus allowing the algorithm to run without constructing any $Q^{(i,*,*)}$ matrices. In this sense, we consider $Q^{(i,*,*)}$ to be pre-computed, thereby separating the cost of its construction from the run-time complexity of the algorithm.

In step 2(b), a QR factorization is used to find the least squares solution of the matrix equation (2.36). The cost of QR factorization grows as the number of subdomains increases (see Section 2.1.5). However, once the factorization has been performed, changes to the right-hand sides of (2.1a) and (2.1b) (i.e., f and ϕ, respectively) do not affect the left-hand side of (2.36). Thus, for a series of problems where only f and ϕ vary, the cost of the QR factorization is only accrued on the first problem, effectively sharing its cost between such problems. Examples of the time saved in such cases are reported in Section 2.2. Further, in the case where ϕ changes while f remains the same, step 1(c) can also be reused, thus starting the algorithm from step 2(c) and saving the cost of applying $\mathscr{G}^{(h)}$ in step 1(c).

Finally, if the type of boundary condition is changed on a given $\Gamma^{(i,j)}$ by changing the piecewise-constant values of α or β on the left-hand side of (2.1b), then the algorithm can begin at step 2(a). However, in practice we do not exploit this case for time savings as step 2(a) is relatively inexpensive to compute.

2.1.5 Complexity

The complexity of the algorithm depends on two main factors: Solving the discrete AP (i.e. applying the operator $\mathscr{G}^{(h)}$) and computing the QR-factorization of $\overline{Q}_N$ from (2.36). Further, the applications of $\mathscr{G}^{(h)}$ include the pre-computed construction of $Q^{(i,*,*)}$ and the run-time steps 1(c) and 3(c) of the algorithm. We emphasize that our algorithm provides the exact solution of the discrete Helmholtz problem, as opposed to the traditional DDMs that are typically iterative. Due to the non-iterative nature of our method, a direct comparison of its complexity to that of the conventional DDMs is poorly defined and not explored in this chapter. We, however, provide a thorough analysis of the complexity of our method as it depends on the various parameters of the discretization.

First, consider applications of $\mathscr{G}^{(h)}$. This operator is only applied to individual subdomains, so let n be the number of grid nodes in one direction in the discretization of an auxiliary domain. Recall from Section 2.1.1.1 the choice of boundary conditions for the $y-$boundaries in (2.2) and the requirement that k be constant on Ω_i. These choices allow the discrete AP to be solved with a combination of a sine-FFT in the $y-$direction and a tridiagonal solver in the $x-$direction, yielding a complexity of $\mathcal{O}(n^2 \log n)$. This is the contribution of one application of $\mathscr{G}^{(h)}$, but $\mathscr{G}^{(h)}$ gets applied many times over the course of the method. In particular, the construction of any $Q^{(i,*,*)}$ requires $8M^*$ applications of $\mathscr{G}^{(h)}$ — one for each column — giving the construction of $Q^{(i,*,*)}$ a complexity of $\mathcal{O}\left(M^* n^2 \log n\right)$. In the worst-case scenario where every subdomain has a unique value of k, the construction of all N of the distinct $Q^{(i,*,*)}$ matrices is $\mathcal{O}\left(NM^* n^2 \log n\right)$. However, it is important to note that the columns of $Q^{(i,*,*)}$ are independent of one another, allowing the construction of $Q^{(i,*,*)}$ to be distributed to a number of parallel processors up to the horizontal dimension of the matrix. Furthermore, $\mathscr{G}^{(h)}$ is also applied twice to every subdomain to construct the right-hand side of (2.34), and an additional application is required in (2.8) to obtain the final approximation. These $3N$ applications contribute $\mathcal{O}(Nn^2 \log n)$ to the overall method complexity.

The cost of the QR-factorization of $\overline{Q}_N$ from (2.36) is the other main contribution to the method's complexity. Assuming that (2.36) is formed by resolving the boundary and transmission conditions as discussed in Section 2.1.2.3, the dimensions of $\overline{Q}_N$ are $N|\gamma| \times 4NM^*$. Note that $|\gamma|$ is roughly proportional to n because γ only contains those nodes closest to the boundary of the subdomain, see Figure 2.4c. The QR-factorization of a matrix depends linearly on the first dimension and quadratically on the second, giving our QR-factorization a complexity of $\mathcal{O}\left((Nn)(NM^*)^2\right)$, or equivalently, $\mathcal{O}\left(N^3 n (M^*)^2\right)$. Further, M^* is typically held constant for a given collection of problems (the selection of M^* is discussed further in Section 2.2), so we consider the cost to be $\mathcal{O}\left(N^3 n\right)$. When the number of subdomains is large, it will become necessary to avoid repeating the QR-factorization when possible as discussed in Section 2.1.4, which leaves us with the $\mathcal{O}\left(N^2 n\right)$ operation of multiplying

$R^{-1}Q^*$ by the source vector $\overline{F}_N$.

Combining the costs of solving the AP and computing the QR-factorization gives the method an overall complexity of $\mathcal{O}\left(N^3 n + N n^2 \log n\right)$. For comparison, consider a simplified situation: let Ω be a square, and let $N = N_d^2$ such that there are N_d subdomains along each side of Ω, allowing the complexity to be rewritten as $\mathcal{O}\left(nN_d^6 + n^2 N_d^2 \log(n)\right)$. Consider solving (2.1) over this Ω with a finite-difference method and without domain decomposition. If the wavenumber is constant and uniform, and the boundary conditions on either the $x-$ or $y-$boundaries are homogeneous Dirichlet conditions, then we can directly use the FFT-based solver mentioned earlier. This domain has nN_d nodes in each direction, so the complexity of this method would be $\mathcal{O}\left((nN_d)^2 \log(nN_d)\right)$. This complexity is better than that of the proposed method. Yet we stress that the FFT-based solver is only applicable in this simplest case, and is inflexible in terms of boundary conditions and variation of the wavenumber. In order to relax the requirements on the wavenumber and boundary conditions, we would need to resort to an LU or similar factorization to invert the finite difference operator with a complexity of $\mathcal{O}\left(((nN_d)^2)^3\right) = \mathcal{O}\left((nN_d)^6\right)$. This approach can capture a wide range of boundary conditions and a variable wavenumber, but it is still ill-suited to cases with piecewise-constant k, as the global solution loses regularity at the interfaces. In these simple cases where all methods apply, the FFT- and LU-based solvers provide lower and upper bounds for the expected performance of our method. However, unlike these two methods, our method extends naturally to more complex domain shapes and boundary conditions.

2.2 Numerical Results

In this section, we present numerical results corroborating the fourth-order convergence of the method, as well as the theoretical computational costs as discussed in Section 2.1.5. For all the test cases, we consider the Helmholtz equation (2.1) where the wavenumber k is constant on any given Ω_i but piecewise constant over Ω. The coefficients α and β from (2.1b) are piecewise constant on each $\Gamma^{(i)}$ such that each is constant on any given subdomain edge, $\Gamma^{(i,j)}$. However, for convenience in presentation, most examples use a uniform boundary condition across the entire $\partial\Omega$.

In all the test cases, we choose the domain Ω such that it can be split into N identical, square subdomains Ω_i, whose interfaces are all full edges. Each Ω_i has side length 2 and every corresponding auxiliary domain is a square with side length 2.2. For consistency, we always let Ω_1 be centered at the origin. Two particular domain shapes that provide convenient and systematic settings for analysis are a long duct and a large square. The duct is a quasi-one dimensional decomposition where Ω_{i+1} extends from Ω_i in the positive $x-$direction, allowing us to directly observe various behaviors of the method with respect to the number of subdomains N. On the other hand, a square domain will be decomposed into $N = N_d^2$ subdomains as discussed in Section 2.1.5, where Ω_1 is again centered at the origin, and acts as the “bottom-left" corner of Ω (see Figure 2.11), with $N-1$ subdomains extending in both the positive $x-$ and $y-$ directions. This domain gives us less direct control over N itself, but it provides a framework to observe the method's performance in the presence of an increasing number of interior cross-points.

All auxiliary problems are solved with the method of difference potentials employing the fourth-order accurate compact finite difference scheme (1.9) on a series of Cartesian grids, starting with $n = 64$ cells uniformly spaced in each direction and progressively doubling n with each refinement. The number of basis functions M^* used in the expansion of $\xi_{\Gamma^{(i,j)}}$ is generally chosen grid-independent [Medvinsky et al., 2012], such that the boundary data are represented to a specified tolerance that is smaller than the error attainable on the given grids. Further increasing M^* offers little to no benefit in the final accuracy of the method as we are still limited by the accuracy of the finite difference scheme. It has even been observed that selecting M^* too large can have adverse effects on the overall accuracy [D. S. Britt et al., 2013], particularly on coarse grids. In this event, it is sufficient to simply reduce M^* for the coarse grids, a practice that we indicate in the relevant results.

There are two kinds of test problems in this section: those with a known exact solution, and those without a known solution. For the test cases with a known solution, the source term and boundary data are derived by substituting the solution into the left-hand side of (2.1a) and (2.1b), respectively. In this case, the error is computed by taking the maximum norm of the difference between the approximated and the exact solution on the grid $\mathbb{M}^+$ (across all subdomains). The convergence rate is then determined by taking the binary logarithm of the ratio of errors on successively refined grids. In general, these test cases either have a uniform wavenumber throughout the domain, or are posed across a small number of subdomains in order to simplify the derivation of an exact solution.

On the other hand, when we want to specify the boundary conditions, source function, or piecewise constant wavenumber, we do not necessarily have an exact solution available and therefore cannot compute the error directly to determine convergence. Instead, we introduce a grid-based metric where we compare the approximations on the shared nodes of successively refined grids. For a grid with $n \times n$ nodes, we denote the corresponding approximation by $u_n^{(h)}$. Because of how the grids are structured, the nodes of the $\frac{n}{2} \times \frac{n}{2}$ grid are a subset of those in the $n \times n$ grid, so we can compute the maximum norm of the difference between these approximations, $\| u_n^{(h)} - u_{\frac{n}{2}}^{(h)} \|_\infty$, on the nodes of $\mathbb{M}^+$ from the $\frac{n}{2} \times \frac{n}{2}$ grid. Similar to the first case, we can then estimate the convergence rate by considering the binary logarithm of the ratio of these maximum norm differences on successive grids. The new convergence metric does not evaluate the actual error. If, however, the discrete solution converges to the continuous one with a certain rate in the proper sense, then this alternative metric will also indicate convergence with the same rate regardless of whether the continuous solution is known or not and so it is convenient to use when the true solution is not available.

All of the computations in this section were performed in MATLAB (ver. R2019a) and used the package Chebfun [Driscoll et al., 2014] to handle all Chebyshev polynomial related operations. The QR-factorizations were performed using MATLAB's built-in 'economy-size' QR-factorization.

2.2.1 Uniform Wavenumber

In order to measure the complexity of the solver, we start with the case of one subdomain (i.e. $\Omega_1 \cong \Omega$). We use the homogeneous test solution $u = e^{i\frac{k}{\sqrt{2}}(x+y)}$ with wavenumber $k = 13$, $M^* = 40$, and Robin boundary conditions defined by $\alpha = 1$ and $\beta = 1$ in (2.1b). The results in Table 2.1 corroborate both the fourth-order convergence rate of the overall method and the computational complexity of solving the discrete AP. As discussed in Section 2.1.5, for a grid with n nodes in each direction, the FFT-based solver should have a complexity of $\mathcal{O}(n^2 \log n)$, which produces the scale factors of approximately 4 in the right-most column of Table 2.1 as n is doubled. Note that Table 2.1 also corroborates the same complexity for the construction of Q from (2.22) (or equivalently, $Q^{(i,*,*)}$ from (2.34)) because the dominating cost of constructing Q is the application of $\mathscr{G}^{(h)}$ for every basis function in the subdomain. Note that, the $\mathscr{G}^{(h)}$ timings will remain approximately constant for any number of subdomains N (up to the available number of processors), with only small increases due to the overhead incurred by parallel communication.

Table 2.1 Grid convergence and time (in seconds) to apply $\mathscr{G}^{(h)}$ to solve the discrete AP for the single subdomain base case. The test solution is $e^{i\frac{k}{\sqrt{2}}(x+y)}$ with $k = 13$, the Robin boundary conditions are defined by $\alpha = \beta = 1$ in (2.1b), and $M^* = 40$.

n	Error	Rate	$\mathscr{G}^{(h)}$ time	Ratio
64	1.05e-03	-	0.0064	-
128	6.42e-05	4.04	0.0083	1.30
256	3.94e-06	4.03	0.042	5.07
512	2.47e-07	4.00	0.186	4.44
1024	1.53e-08	4.01	0.824	4.44
2048	9.89e-10	3.95	3.430	4.17

Next, we consider cases where Ω can be decomposed into two or more subdomains, under the simplifying assumption that k is uniformly constant throughout Ω, sharing the same value on every Ω_i. This case is the simplest to consider because every Ω_i will use the same $Q^{(i,*,*)}$, removing the need to compute a new $Q^{(i,*,*)}$ for every possible value of k in a given problem. Tables 2.2 – 2.5 display the grid convergence for several examples that were derived from known test solutions. Note that for each of these tables, changing the test solution only affects the source and boundary data, which means $\overline{Q}_N$ from (2.36) and its QR-factorization remain the same for all three test problems. Hence, for each of Tables 2.2 – 2.5, the QR-factorization is only computed once (during the first case) and is reused for the second and third cases, allowing those problems to be solved at a reduced cost.

Table 2.2 shows the grid convergence of the case with two subdomains, which uses the example domain given in Figure 1.6 with basic Robin boundary conditions defined by $\alpha = 1$ and $\beta = 1$ in (2.1b). By comparing the grid convergence of the first test case in Table 2.2 to that of Table 2.1, we can see that including the domain decomposition does not affect the convergence rate of the method, and also has very little effect on the error itself. The second and third test solutions are both inhomogeneous, and clearly still converge with the designed rate of convergence.

Table 2.2 Grid convergence for three test solutions over a two-subdomain Ω where Ω_1 is centered at the origin and the interface with Ω_2 is at $x = 1$. A uniform wavenumber $k = 13$ is used in both subdomains, $M^* = 40$, and the Robin boundary condition is uniformly defined by $\alpha = \beta = 1$ in (2.1b). Errors marked with * are computed with $M^* = 20$.

	$e^{i\frac{k}{\sqrt{2}}(x+y)}$		$e^{\frac{-1}{1-(x^2+y^2)}}$		$\sin^4(\pi x)\sin(\pi y)$	
n	Error	Rate	Error	Rate	Error	Rate
64	8.52e-03	-	7.93e-04*	-	1.37e-03	-
128	5.15e-04	4.05	3.16e-06*	7.97	8.32e-05	4.04
256	3.12e-05	4.05	2.18e-07	3.86	5.13e-06	4.02
512	1.91e-06	4.03	1.34e-08	4.03	3.19e-07	4.01
1024	1.20e-07	4.00	8.31e-10	4.01	2.00e-08	4.00
2048	7.37e-09	4.02	5.18e-11	4.00	1.26e-09	3.99

In Table 2.3, we use the exact same test solutions, wavenumber, and boundary conditions as in Table 2.2, but on a larger scale with $N = 24$ subdomains extending in the positive x—direction from Ω_1. On this larger scale, we can directly observe the $\mathcal{O}(N^3 n)$ complexity of the QR-factorization with respect to n, which was explained in Section 2.1.5 (see Table 2.6 for the complexity with respect to N). In the first two columns of Table 2.3, we see that as the grid dimension n is doubled, the time for the QR-factorization is approximately doubled as well.

Table 2.3 Grid convergence and QR-factorization timing for three test solutions where Ω is a long duct of $N = 24$ subdomains. Ω_1 is centered at the origin and each subsequent Ω_i is attached horizontally in the positive x—direction. A uniform wavenumber $k = 13$ is used in all subdomains, $M^* = 40$, and the Robin boundary condition is uniformly defined by $\alpha = \beta = 1$ in (2.1b). The ratios of QR times demonstrates linear complexity with respect to the grid dimension n. Errors marked with * were computed with $M^* = 20$. Note that the QR-factorization does not need repeated for the * cases, as their factorizations can be extracted directly from the existing factorization in each case.

			$e^{i\frac{k}{\sqrt{2}}(x+y)}$		$e^{\frac{-1}{1-(x^2+y^2)}}$		$\sin^4(\pi x)\sin(\pi y)$	
n	QR Time	Ratio	Error	Rate	Error	Rate	Error	Rate
64	3.23	-	3.89e-02	-	8.34e-04*	-	1.07e-03	-
128	6.58	2.00	8.08e-04	5.59	4.29e-06*	7.60	6.38e-05	4.07
256	13.50	2.05	4.85e-05	4.06	6.10e-07	2.81	3.95e-06	4.01
512	28.91	2.14	3.01e-06	4.01	3.80e-08	4.00	2.46e-07	4.01
1024	60.13	2.08	1.89e-07	4.00	2.38e-09	4.00	1.54e-08	4.00
2048	108.77	1.81	1.39e-08	3.77	1.51e-10	3.98	1.02e-09	3.91

Table 2.4 Grid convergence for three test solutions where Ω is a large square comprising of 3 subdomains in each direction, and the bottom left subdomain is centered at the origin. A uniform wavenumber $k = 13$ is used in all subdomains, $M^* = 40$, and the Robin boundary condition is uniformly defined by $\alpha = \beta = 1$ in (2.1b). Errors marked with * are computed with $M^* = 20$.

	$e^{i\frac{k}{\sqrt{2}}(x+y)}$		$e^{\frac{-1}{1-(x^2+y^2)}}$		$\sin^4(\pi x)\sin(\pi y)$	
n	Error	Rate	Error	Rate	Error	Rate
64	9.08e-03	-	8.63e-04*	-	1.86e-03	-
128	5.76e-04	3.98	4.76e-06*	7.50	1.12e-04	4.06
256	3.63e-05	3.99	1.64e-07	4.86	6.92e-06	4.01
512	2.27e-06	4.00	1.02e-08	4.01	4.30e-07	4.01
1024	1.42e-07	4.00	6.39e-10	4.00	2.70e-08	4.00
2048	8.83e-09	4.01	4.01e-11	3.99	1.69e-09	3.99

The case of a large square domain is reported in Table 2.4, with the same three test solutions as previous tables. In this case, there is a subdomain that is entirely interior and therefore has no boundary condition given, as well as numerous cross-points where more than two subdomains meet at a single point. As can be seen in Table 2.4, the errors and convergence rate are unaffected by the presence of an interior subdomain and cross-points in this simple case, while more extreme cases are presented in Tables 2.10 and 2.11 in Section 2.2.2.

Table 2.5 Grid convergence for three test solutions over two subdomains with mixed boundary conditions as indicated in Figure 2.7. A uniform wavenumber $k = 13$ is used in both subdomains, and $M^* = 40$. Errors marked with * are computed with $M^* = 20$.

	$e^{i\frac{k}{\sqrt{2}}(x+y)}$		$e^{\frac{-1}{1-(x^2+y^2)}}$		$\sin^4(\pi x)\sin(\pi y)$	
n	Error	Rate	Error	Rate	Error	Rate
64	1.73e-03	-	2.08e-04*	-	9.61e-04	-
128	1.05e-04	4.04	3.49e-06*	5.90	5.86e-05	4.04
256	6.55e-06	4.01	1.56e-07	4.49	3.62e-06	4.02
512	4.10e-07	4.00	9.69e-09	4.01	2.25e-07	4.01
1024	2.56e-08	4.00	6.06e-10	4.00	1.41e-08	3.99
2048	1.60e-09	4.00	3.79e-11	4.00	8.77e-10	4.01

Table 2.5 displays the grid convergence for a problem with mixed boundary conditions (see also Figure 2.7), showing that the method is robust enough to handle such boundary conditions without affecting its convergence rate. The domain, wavenumber, and test solutions in Table 2.5 are the same as in Table 2.2, saving the cost of two applications of $\mathscr{G}^{(h)}$ (per subdomain) because F in the right-hand side of (2.34) does not need to be computed. As the type of boundary condition changed (i.e., α or β in (2.1b) changed) we need to recompute the QR-factorization for the first test solution,

reusing it for the second and third test solutions.

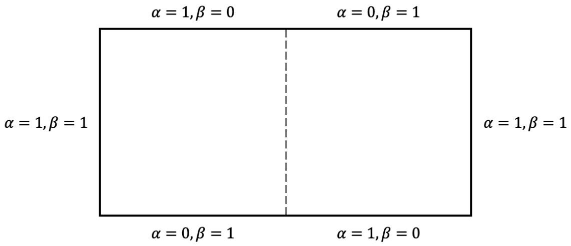


Figure 2.7 The mixed boundary condition for Table 2.5. The coefficients α and β are defined separately for each exterior edge of the subdomains.

In Table 2.6, we can see how the timing for the QR-factorization grows with respect to the number of subdomains, N. Recall from Section 2.1.5 that the complexity of the QR-factorization should be $\mathcal{O}\left(N^3 n\right)$, so as N is doubled we would expect to see the execution time of the QR-factorization increase by a factor of 8. However, as can be seen in Table 2.6, the execution time of the QR-factorization actually scales slower than its theoretical complexity would suggest, at least for the sizes of problems we were able to test.

Table 2.6 Timings for the QR-factorization (in seconds) on the 2048 × 2048 grid. Ω is a long duct of N subdomains, and the test solution is $u = e^{i\frac{k}{\sqrt{2}}(x+y)}$ with $M^* = 40$ and $k = 13$ in all subdomains. The Robin boundary condition is defined by $\alpha = \beta = 1$ in (2.1b).

N	QR time	Ratio
1	0.064	-
2	0.29	4.61
4	1.66	5.65
8	11.72	7.04
16	61.89	5.28
32	362.42	5.86

2.2.2 Piecewise-Constant Wavenumber

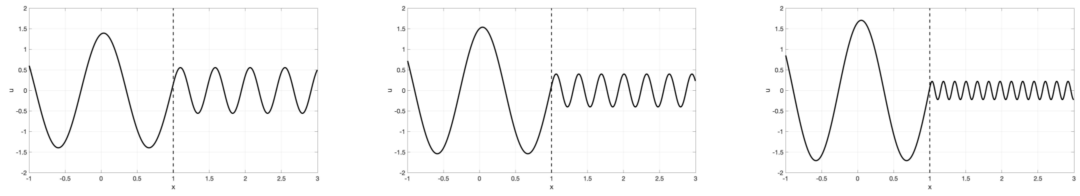
We now focus on cases where k is piecewise-constant over Ω (with constant value k_i over any Ω_i). New $Q^{(i,*,*)}$ matrices are needed for any new values of k_i, but recall that we only need to compute $Q^{(i,*,*)}$ once for each unique k_i. For the results in this section, it is assumed that any necessary $Q^{(i,*,*)}$ matrices were appropriately computed ahead of time.

Tables 2.7 and 2.8 show the grid convergence for the two-subdomain and four-subdomain cases, respectively. In both cases, the test solution is obtained by considering an incident wave, e^{ik_1x} in

Ω_1, and deriving the corresponding reflected and transmitted waves by enforcing continuity of the function and its normal derivative at each interface. This derivation can be found in Section 2.3. Table 2.7 shows the results of taking $k_1 = 5$ and allowing jumps to k_2, varying between 13, 20, and 40, with cross-sections of each of these solutions plotted in Figure 2.8. The method maintains its fourth-order rate of convergence, even on the largest jump from 5 to 40. It is worth pointing out here that as k_2 is increased, we need to increase M^* because more oscillatory solutions will require more basis functions to maintain high-order accuracy. In cases where $M^* \geq 50$, this causes a loss of accuracy on the coarsest grid, so M^* is reduced only for the $n = 64$ grid in the relevant test cases. The four-subdomain case presented in Table 2.8 was derived similar to the two-subdomain case, but only for one test solution. The values of k on each subdomain in this example are $k_1 = 3$, $k_2 = 5$, $k_3 = 13$, and $k_4 = 20$, and the solution is plotted in Figure 2.9. Even with four unique wavenumbers, it can be seen in Table 2.8 that the method still has fourth-order convergence.

Table 2.7 Grid convergence for the two-subdomain test case with the incident wave $u = e^{ik_1 x}$ and Dirichlet boundary conditions, as plotted in Figure 2.8. The jump in wavenumber goes from $k_1 = 5$ to the indicated value of k_2, and M^* is chosen separately for each case to ensure accuracy beyond that obtained on the finest grid. Errors marked with a * were computed with $M^* = 30$.

	$k_1 = 5$	$k_2 = 13$	$k_1 = 5$	$k_2 = 20$	$k_1 = 5$	$k_2 = 40$
	$M^* = 40$		$M^* = 50$		$M^* = 60$	
n	Error	Rate	Error	Rate	Error	Rate
64	3.23e-02	-	8.81e-02*	-	1.27e+01*	-
128	1.98e-03	4.03	5.13e-03	4.10	1.08e-01	6.88
256	1.21e-04	4.04	3.13e-04	4.03	6.51e-03	4.06
512	7.59e-06	3.99	1.96e-05	4.00	4.03e-04	4.01
1024	4.69e-07	4.02	1.22e-06	4.01	2.52e-05	4.00
2048	3.09e-08	3.92	7.78e-08	3.97	1.57e-06	4.01



(a) $k_1 = 5$, $k_2 = 13$ **(b)** $k_1 = 5$, $k_2 = 20$ **(c)** $k_1 = 5$, $k_2 = 40$

Figure 2.8 Real part of the test solutions from Table 2.7. These solutions have no dependence on y, so cross-sections in the x−direction are plotted. In each plot, it is clear that the frequency of the plane-wave changes at the interface.

Table 2.8 Grid convergence for $u = e^{ikx}$ over four subdomains with Dirichlet boundary conditions, and wavenumbers $k_1 = 3$, $k_2 = 5$, $k_3 = 13$, and $k_4 = 20$. Ω_1 is centered at the origin, and each subdomain extends in the positive x—direction, with $M^* = 50$. The error marked with a * was computed with $M^* = 30$.

n	Error	Rate
64	2.02e-02*	-
128	1.21e-03	4.08
256	7.38e-05	4.03
512	4.61e-06	4.00
1024	2.87e-07	4.01
2048	2.05e-08	3.80

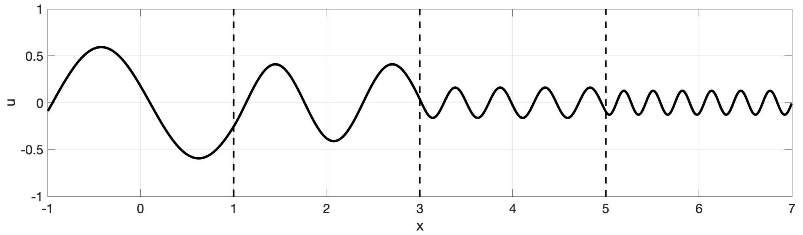


Figure 2.9 Real part of the test solution from Table 2.8. The solution has no dependence on y, so a cross-section is plotted. Each Ω_i has a distinct wavenumber k_i, with $k_1 = 3$, $k_2 = 5$, $k_3 = 13$, and $k_4 = 20$. Moving in the positive x—direction, the wavenumber and frequency increase, while the amplitude decreases.

Additionally, we point out the increase in error as k_2 increases in Table 2.7. We attribute this increase to the pollution effect [Bayliss et al., 1985], because it appears consistent with our additional observations of the pollution effect for problems with uniform wavenumbers (no jumps) that are comparable to k_2. We therefore conclude that the method is not inherently sensitive to discontinuities in the wavenumber.

As we allow the test cases to become more complex in geometry and wavenumber distribution, it becomes more difficult to obtain analytic test solutions. Instead, we specify the boundary and source data directly, and calculate errors on shared nodes between subsequent resolutions of the grid. For simplicity, the source function is taken to be a "bump" function:

$$f(x,y) = \begin{cases} \exp\left(\frac{-1}{\frac{1}{4}-(x^2+y^2)}\right) & x^2+y^2 < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.37}$$

In Table 2.9, we present an example of a long duct of $N = 16$ subdomains with a change in wavenumber at every interface, alternating between $k = 5$ and $k = 40$ (depicted in Figure 2.10). The three cases in Table 2.9 represent homogeneous Dirichlet, Neumann, and Robin ($\alpha = \beta = 1$) boundary conditions, respectively, and show that the method maintains its design rate of convergence for all three types of boundary conditions.

In Table 2.10, the domain is a large square decomposed into $N = N_d^2$ (cf. Section 2.1.5) sub-

Table 2.9 Grid convergence on the duct of $N = 16$ subdomains depicted in Figure 2.10, various homogeneous boundary conditions, and the source function from (2.37). The wavenumbers alternate between $k = 5$ and $k = 40$, and $M^* = 60$.

	Dirichlet		Neumann		Robin	
n	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	Rate	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	Rate	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	Rate
256	1.51e-03	-	5.79e-04	-	3.50e-04	-
512	1.08e-04	3.81	7.54e-06	6.26	3.57e-06	6.62
1024	6.76e-06	3.99	4.62e-07	4.03	2.19e-07	4.03
2048	4.22e-07	4.00	2.90e-08	3.99	1.38e-08	3.99

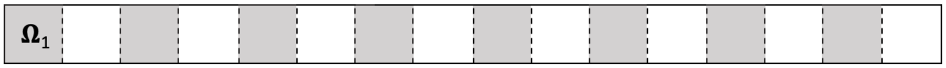


Figure 2.10 The decomposition used to compute Table 2.9 where the wavenumbers are alternating in each subdomain. Ω_1 is indicated in the left-most subdomain with each subsequent subdomain being attached in the positive x−direction. Wavenumbers are assigned as $k = 5$ for gray subdomains and $k = 40$ in white subdomains.

domains, where the piecewise constant values of the wavenumber are defined in a checkerboard pattern with $k = 5$ and $k = 40$ as in Figure 2.11. These examples combine the qualitative aspects of Tables 2.4 and 2.9, containing cross-points as well as a changing wavenumber at every interface (now in both the x− and y−directions). We emphasize that no special considerations were given to internal or boundary cross-points, yet the method's convergence does not suffer from their presence.

Table 2.10 Grid convergence on a large square decomposed into $N = N_d^2$ subdomains and a checkerboard pattern for its wavenumber as depicted in Figure 2.11. There are homogeneous Dirichlet boundary conditions and the source function is (2.37). The checkerboard wavenumbers are $k = 5$ (gray) and $k = 40$ (white), with $M^* = 60$.

	$N = 16$		$N = 25$		$N = 36$	
n	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	Rate	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	Rate	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	Rate
256	8.54e-04	-	6.48e-03	-	4.31e-04	-
512	6.90e-05	3.63	7.45e-04	3.12	1.76e-05	4.61
1024	4.36e-06	3.98	4.90e-05	3.93	1.09e-06	4.01
2048	2.73e-07	4.00	3.06e-06	4.00	6.79e-08	4.01

Table 2.11 shows the final example, which returns to the case of a square domain decomposed into 3 × 3 subdomains, but with wavenumbers assigned as in Figure 2.12. In contrast to the configurations of Figures 2.10 and 2.11, this example uses a different wavenumber in each subdomain (similar to Figure 2.9). This is the costliest test case for the method, as $Q^{(i,*,*)}$ is different for every Ω_i.

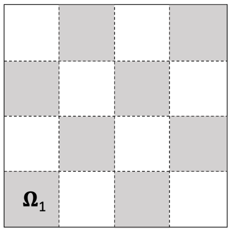
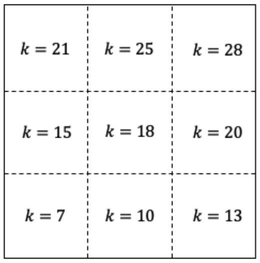


Figure 2.11 The 4×4 decomposition used in the first case of Table 2.10 where the wavenumbers are assigned in a checkerboard pattern, with $k = 5$ in the gray subdomains and $k = 40$ in the white subdomains.



$k = 21$	$k = 25$	$k = 28$
$k = 15$	$k = 18$	$k = 20$
$k = 7$	$k = 10$	$k = 13$

Figure 2.12 Piecewise constant values of the wavenumber k for the example computed in Table 2.11.

However, we can still use those $Q^{(i,*,*)}$ matrices computed for $k = 5, 13$, and 20 that were used for earlier examples. In Table 2.11, fourth-order convergence is clearly observed.

Table 2.11 Grid convergence on a square domain decomposed into $N = 9$ subdomains with homogeneous Dirichlet boundary conditions and the source function from (2.37). Each subdomain has a unique wavenumber as depicted in Figure 2.12, and $M^* = 60$.

n	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	Rate
256	9.04e-05	-
512	3.14e-06	4.85
1024	1.89e-07	4.05
2048	1.20e-08	3.98

2.2.3 Further Studies

Thus far, we have focused on domains that are either squares or ducts, making it straightforward to generate solutions that have no singularities and can ensure the design rate of convergence for the method. Moving beyond simple square and duct decompositions will often introduce reentrant corners, an example of which is shown in the third plot of Figure 2.6. Reentrant corners may cause a solution to develop a singularity on the boundary, which in turn can cause the method to lose its design rate of convergence. In order to observe this phenomenon, we introduce the “Block L”

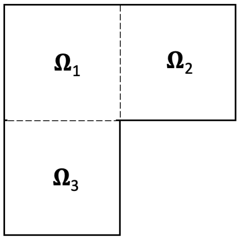


Figure 2.13 The "Block L" domain decomposition used in Table 2.12. Ω_1 is centered over the origin, with interfaces to Ω_2 and Ω_3 at $x = 1$ and $y = -1$, respectively. This configuration creates a reentrant corner at the point $(x, y) = (1, -1)$.

domain in Figure 2.13 which contains one reentrant corner, and demonstrate the convergence obtained by our method under two different types of solutions.

Table 2.12 shows the grid convergence over the Block L domain. The first case is derived from the exact solution $u = e^{i\frac{k}{\sqrt{2}}(x+y)}$, and the second case uses a homogeneous Dirichlet boundary condition with the source function (2.37). In the first case, the test solution contains no singularities, and as expected Table 2.12 reflects the design rate of convergence. This indicates that the reentrant corner itself is not an issue for our method. However, the solution in the second case develops a singularity, and the breakdown in the convergence rate in Table 2.12 is indicative of that. For Case 1, we are able to compute the error directly with the known test solution, but for Case 2 we compute the error on successively refined grids as described earlier.

The breakdown of convergence in the second case is a result of the solution's own singularity, rather than a shortcoming of the method. The resolution of singularities at reentrant corners with the MDP has been explored in [Magura et al., 2017]. The method can also handle general shaped subdomains but these fall outside the scope of this dissertation.

Table 2.12 Grid convergence for the Block L case from Figure 2.13 with Dirichlet boundary conditions and uniform wavenumber $k = 13$, with $M^* = 40$. In Case 1, the boundary and source data are derived from the plane-wave $u = e^{i\frac{k}{\sqrt{2}}(x+y)}$, which results in fourth order convergence. In Case 2, the boundary data is zero, and the source function is given in (2.37). Case 2 develops a singularity at the reentrant corner and breaks down the convergence.

	Case 1		Case 2	
n	Error	Rate	$\Vert u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\Vert_\infty$	Rate
64	9.58e-04	-	-	-
128	6.03e-05	3.99	4.08e-06	-
256	3.76e-06	4.00	1.98e-06	1.04
512	2.35e-07	4.00	1.09e-06	0.86
1024	1.47e-08	4.00	1.02e-06	0.10
2048	9.80e-10	3.91	4.70e-07	1.12

2.3 Supplemental Derivation of a Test Solution

Consider a domain $\Omega \subset \mathbb{R}^2$ split into two subdomains, Ω_1 and Ω_2, as in Figure 2.14. Let each subdomain have its own corresponding wavenumber, k_1 or k_2, which we will assume is constant for simplicity in this derivation. We seek a function $u \in C^1(\Omega)$ that incorporates the reflected and transmitted parts of an incident wave that starts in Ω_1 and propagates toward the interface, located at $x = 0$ for simplicity.

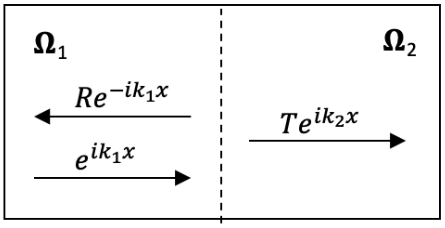


Figure 2.14 The setup for deriving the reflected and transmitted parts of a one-dimensional incident wave hitting an interface in a domain composed of two subdomains.

Let $u_1 = e^{ik_1x}$ be the incident wave in Ω_1. When the wavenumber changes at the interface $x = 0$, u_1 is partially reflected back into Ω_1 and partially transmitted through to Ω_2. The reflected wave has some amplitude R and travels in the opposite direction of u_1, giving us $u_2 = Re^{-ik_1x}$. The transmitted part, on the other hand, will have its own amplitude T, traveling in the same direction as u_1 and with the wavenumber k_2, giving $u_3 = Te^{ik_2x}$. This allows us to write the function u as:

$$u(x,y) = \begin{cases} e^{ik_1x} + Re^{-ik_1x}, & x \leqslant 0 \\ Te^{ik_2x}, & x \geqslant 0 \end{cases}. \tag{2.38}$$

The condition to enforce continuity of the function at the interface is

$$\begin{aligned} e^{ik_1x}|_{x=0} + Re^{-ik_1x}|_{x=0} = Te^{ik_2x}|_{x=0} \\ \implies 1 + R = T \end{aligned} \tag{2.39}$$

and for continuity of the derivative we get

$$
\begin{aligned}
\left(\frac{\partial}{\partial x}e^{ik_1x}\right)|_{x=0}+\left(\frac{\partial}{\partial x}Re^{-ik_1x}\right)|_{x=0}&=\left(\frac{\partial}{\partial x}Te^{ik_2x}\right)|_{x=0}\\
\Longrightarrow \left(ik_1e^{ik_1x}\right)|_{x=0}-\left(ik_1Re^{-ik_1x}\right)|_{x=0}&=\left(ik_2Te^{ik_2x}\right)|_{x=0}\\
\Longrightarrow ik_1-ik_1R&=ik_2T\\
\Longrightarrow k_1(1-R)&=k_2T\\
\Longrightarrow 1-R&=\frac{k_2}{k_1}T. \qquad (2.40)
\end{aligned}
$$

By combining (2.39) and (2.40), we can solve for R and T to get

$$R=\frac{k_2}{2k_1}-\frac{1}{2}, \qquad T=\frac{k_2}{2k_1}+\frac{1}{2}. \qquad (2.41)$$

The values of R and T from (2.41) can be plugged into (2.38) to obtain the function $u(x,y)$, defined across Ω.

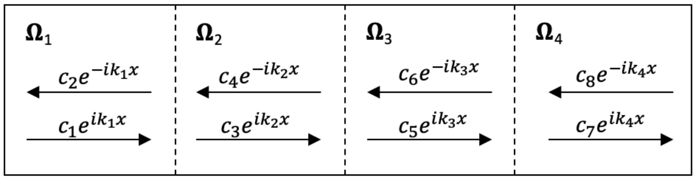


Figure 2.15 The setup for deriving the reflected and transmitted parts of a one-dimensional incident wave hitting an interface in a domain composed of four subdomains.

For a larger case with four subdomains, consider the scenario depicted in Figure 2.15, with interfaces at $x = x_1$, x_2, and x_3. This scenario is a direct extension of the two-subdomain case, and we can obtain a linear system by similarly enforcing continuity of the function and its derivative at each interface. For example, at the interface between Ω_1 and Ω_2 (i.e. $x = x_1$) enforcing continuity of the function itself yields:

$$c_1e^{ik_1x_1}+c_2e^{-ik_1x_1}=c_3e^{ik_2x_1}+c_4e^{-ik_2x_1}$$

which can be rewritten as

$$c_1e^{ik_1x_1}+c_2e^{-ik_1x_1}-c_3e^{ik_2x_1}-c_4e^{-ik_2x_1}=0.$$

By including the corresponding conditions for both the function and its derivative at all three interfaces, we get the following system of equations:

$$\begin{aligned}
c_1 e^{ik_1x_1} + c_2 e^{-ik_1x_1} - c_3 e^{ik_2x_1} - c_4 e^{-ik_2x_1} &= 0\\
c_3 e^{ik_2x_2} + c_4 e^{-ik_2x_2} - c_5 e^{ik_3x_2} - c_6 e^{-ik_3x_2} &= 0\\
c_5 e^{ik_3x_3} + c_6 e^{-ik_3x_3} - c_7 e^{ik_4x_3} - c_8 e^{-ik_4x_3} &= 0\\
k_1 c_1 e^{ik_1x_1} - k_1 c_2 e^{-ik_1x_1} - k_2 c_3 e^{ik_2x_1} + k_2 c_4 e^{-ik_2x_1} &= 0\\
k_2 c_3 e^{ik_2x_2} - k_2 c_4 e^{-ik_2x_2} - k_3 c_5 e^{ik_3x_2} + k_3 c_6 e^{-ik_3x_2} &= 0\\
k_3 c_5 e^{ik_3x_3} - k_3 c_6 e^{-ik_3x_3} - k_4 c_7 e^{ik_4x_3} + k_4 c_8 e^{-ik_4x_3} &= 0.
\end{aligned}$$

Note that this only provides six equations for eight unknowns. As in the two-subdomain case, we can pick one of the waves to be the incident wave, and choose to set its amplitude to 1 for simplicity, so we can directly choose $c_1 = 1$. Further, the boundary of Ω is not reflective, which means that $c_7 e^{ik_4x}$ does not reflect upon reaching the right boundary, leaving $c_8 = 0$. This leaves six equations for six unknowns, which allows this problem to be solved uniquely.

The same process can be applied to the $N-$subdomain case. If we let the incident wave to be given in Ω_1, then its reflection back into Ω_1 has an undetermined amplitude. For Ω_2 through Ω_{N-1}, there are two waves traveling in opposite directions for which the amplitudes are also undetermined. Finally, there is no reflected wave in Ω_N, so there is only one amplitude to solve for, yielding a total of $1+2(N-2)+1 = 2N-2$ unknowns. This scenario contains $N-1$ interfaces, and each interface has two conditions: continuity of the function and continuity of the derivative. These conditions yield $2(N-1) = 2N-2$ equations, allowing us to solve for the $2N-2$ unknowns.

2.4 Conclusions

We have adapted the Method of Difference Potentials to solve a non-overlapping Domain Decomposition formulation for the Helmholtz equation. After solving for the boundary information along all interfaces, the direct solves for all subproblems can be distributed and performed concurrently. Further, the formulation is convenient for handling piecewise-constant wavenumbers, as well as mixed boundary conditions. Numerical results corroborate the fourth-order convergence rate of the method in numerous situations, most notably for decompositions with cross-points and for transmission problems with a large jump in the wavenumber. Our formulation also allowed us to demonstrate different behaviors of the method, such as its performance on solutions with singularities and the method's complexity with respect to the number of subdomains or grid dimension.

Once the boundary/interface data have been obtained for the subdomains, only one direct solve is required per subdomain, and this set of direct solves can be parallelized on a number of processors up to the number of subdomains. However, this comes at the cost of requiring the QR-factorization of a large system. Even though the QR-factorization scales slower than its theoretical complexity would suggest, for cases with more than $N = 8$ subdomains the QR-factorization already outweighs the final PDE solves in cost. However, once the QR-factorization has been computed, new problems with variations in the source and boundary data can be solved at a reduced cost by simply reusing the computed QR-factorization. The framework that is laid out in this Chapter can be adapted and extended in numerous ways to broaden the applicability of the method. Briefly mentioned in Section 2.1.2.2, interface conditions of the form (2.31) can be implemented trivially along every interface to account for more complex properties of the solution there, such as jumps in the normal derivative or solution itself. The method can be generalized to account for a smoothly varying wavenumber $k = k(x, y)$ in each subdomain, although it may reduce the efficiency as the FFT-based solver would no longer be applicable. Base subdomains of a different shape could be included, such as those with piecewise-curvilinear boundaries, in order to account for more complex geometries.

In this chapter, we have built a domain decomposition framework from a fundamental building block. In Chapter 3 we implement that framework in conjunction with a new type of building block. This new building block permits simulations over domains with more interesting properties (e.g. refraction across circular rods rather than just across linear interfaces). The implementation of absorbing boundary conditions for these subdomains is also described. Together, the new building block and absorbing boundary conditions allow us to simulate a particular kind of waveguide — a photonic crystal ring resonator — the structure of which contains periodicities and material discontinuities that lend themselves naturally to our method.

CHAPTER

3

COMPOSITE SUBDOMAINS FOR PCRR SIMULATIONS

In addition to the empty square building block (subdomain) described in Section 2.1.1, the decomposition of a PCRR (Figure 1.3) also requires a building block that consists of a square with an interior scattering object. This building block should fit into the decomposition framework described in Chapter 2, allowing edges to be implemented both as exterior edges (boundary conditions) or interior edges (interface conditions). The focus of this chapter is to describe the inclusion of interface conditions along the scattering object inside of this new building block, as well as the implementation of absorbing boundary conditions (Section 1.2.2) at exterior boundary edges. Together, these two additions to the DDM framework allow the method to be used in simulating traditional wave propagation problems that rely on scattering mechanics and outgoing wave propagation.

For the purposes of this chapter, consider again the inhomogeneous Helmholtz equation posed over a domain Ω with boundary $\partial\Omega$, but with a general boundary condition operator $\mathscr{B}u \overset{\text{def}}{=} \alpha u + \beta \frac{\partial u}{\partial n}$:

$$\Delta u + k^2 u = f, \quad \text{in } \Omega \tag{3.1a}$$

$$\mathscr{B}u = \phi, \quad \text{on } \partial\Omega. \tag{3.1b}$$

The wavenumber is still assumed to be constant for a given subproblem. However, whereas Chapter 2 considered wavenumbers that changed between squares, this chapter focuses on the case where the wavenumber changes across the rod, causing refraction of the signal.

This chapter is adapted from [North et al., n.d.] and begins with Section 3.1 which gives the formulation of the Helmholtz equation over the decompositions being discussed in this chapter.

Section 3.2 is an analogue to Section 2.1, introducing and notating the MDP in the context of this rod subdomain. The numerical results of Section 3.3 are split into two sets of results: those that quantify the performance of the method in simplified scenarios (Section 3.3.1) and those that serve as more practical simulations of the expected physics in a PCRR (Section 3.3.2).

3.1 Decomposition Formulation

The decomposition of the PCRR relies on two fundamental building blocks: an empty square and a square that contains a circle (Figure 1.3). The empty square building block can be visualized in Figure 1.6, where a larger (rectangular) domain Ω is split into two identical subdomains Ω_1 and Ω_2 separated by an interface Σ. The formulation of the Helmholtz BVP (3.1) over the two empty subdomains can then be expressed as

$$\begin{cases} \Delta u_1 + k_1^2 u_1 = f, & \text{in } \Omega_1 \\ \mathscr{B} u_1 = \phi, & \text{on } \overline{\Omega}_1 \cap \partial\Omega \end{cases} \qquad \begin{cases} \Delta u_2 + k_2^2 u_2 = f, & \text{in } \Omega_2 \\ \mathscr{B} u_2 = \phi, & \text{on } \overline{\Omega}_2 \cap \partial\Omega \end{cases} \tag{3.2a}$$

$$\begin{cases} u_1 = u_2, & \text{on } \Sigma \\ \frac{\partial u_1}{\partial \mathbf{n_1}} = -\frac{\partial u_2}{\partial \mathbf{n_2}}, & \text{on } \Sigma \end{cases} \tag{3.2b}$$

where the constant wavenumbers k_1 and k_2 characterize the material properties in Ω_1 and Ω_2. As discussed in Section 1.2.1, subproblems (3.2a) only have boundary conditions defined along their exterior edges, leaving them underdetermined. Thus we include the two conditions of (3.2b) to provide the necessary conditions along the interface Σ to ensure the solution of (3.2) on each subdomain matches that of (3.1), i.e.,

$$\begin{cases} u = u_1 & \text{in } \Omega_1, \\ u = u_2 & \text{in } \Omega_2. \end{cases}$$

Although other transmission conditions can be used in place of (3.2b), we note that the solutions obtained in the respective subdomains would no longer match their corresponding parts of the global solution to (3.1).

The second building block, the square that contains a circle (henceforth referred to as the "rod subdomain") is itself treated as a decomposition into two pieces: the circle itself and the rest of the square outside of the circle (see Figure 3.1). In this decomposition, the interface Σ is the circle. Physically, the rod subdomain is used to model parts of the PCRR where the scattering rods exist and create material discontinuities. Therefore, when reformulating (3.1) over the rod subdomain Ω_j

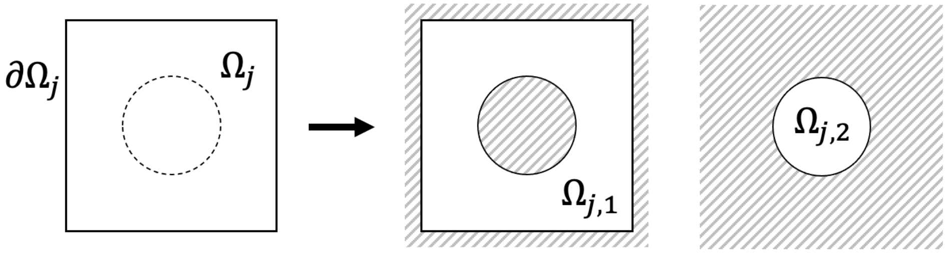


Figure 3.1 Decomposition of the rod subdomain into two of its own subdomains such that $\Omega_j = \Omega_{j,1} \cup \overline{\Omega}_{j,2}$. The dotted circle is not part of $\partial\Omega_j$.

we allow k to change as it crosses the circle, i.e.

$$\begin{cases} \Delta u_1 + k_1^2 u_1 = f & \text{in } \Omega_{j,1} \\ \mathscr{B} u_1 = \phi & \text{on } \partial\Omega_j \end{cases} \tag{3.3a}$$

$$\begin{cases} \Delta u_2 + k_2^2 u_2 = 0 & \text{in } \Omega_{j,2} \end{cases} \tag{3.3b}$$

$$\begin{cases} u_1 = u_2, & \text{on } \overline{\Omega}_{j,1} \cap \overline{\Omega}_{j,2} \\ \frac{\partial u_1}{\partial \mathbf{n_1}} = -\frac{\partial u_2}{\partial \mathbf{n_2}}, & \text{on } \overline{\Omega}_{j,1} \cap \overline{\Omega}_{j,2}. \end{cases} \tag{3.3c}$$

Note that, the subproblem (3.3b) does not have a boundary condition explicitly associated with it, and that the entirety of the boundary condition (3.1b) is associated with (3.3a). Instead, the boundary information for (3.3b) is provided solely by the transmission conditions (3.3c). If $k_1 = k_2$, this subproblem becomes mathematically equivalent to the empty subdomain case (see Section 2.1.1 or [North et al., 2022]), so we generally assume $k_1 \neq k_2$.

For decompositions that incorporate both kinds of subdomains, we follow the notation introduced in Figure 3.2. Interfaces between two separate squares Ω_i and Ω_j are labeled $\Sigma_{i,j}$. Thus, $\Sigma_{i,j}$ and $\Sigma_{j,i}$ are equivalent and can be used interchangeably. Within a rod subdomain Ω_j, the circular interface is simply denoted Σ_j. The decomposition of Ω_j remains consistent with Figure 3.1, where $\Omega_j = \Omega_{j,1} \cup \overline{\Omega}_{j,2}$.

As an example, the following is a breakdown of how (3.1) could be formulated across the domain

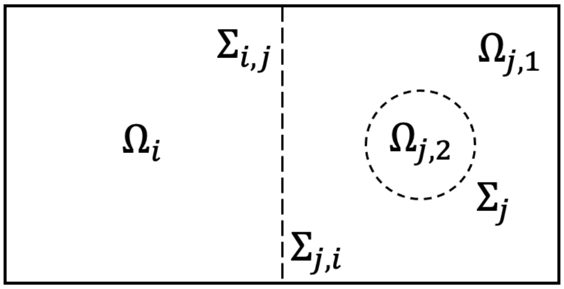


Figure 3.2 Example of how various parts of a composite domain are labeled when considered as part of the same decomposition.

$\Omega = \Omega_i \cup \Omega_j$ depicted in Figure 3.2:

$$\begin{gathered}
\begin{cases} \Delta u_i + k_i^2 u_i = f, & \text{in } \Omega_i \\ \mathscr{B} u_i = \phi, & \text{on } \partial\Omega \cap \overline{\Omega}_i \end{cases} \\
\begin{cases} \Delta u_{j,1} + k_{j,1}^2 u_{j,1} = f, & \text{in } \Omega_{j,1} \\ \mathscr{B} u_{j,1} = \phi, & \text{on } \partial\Omega \cap \overline{\Omega}_{j,1} \end{cases} \\
\begin{cases} \Delta u_{j,2} + k_{j,2}^2 u_{j,2} = f, & \text{in } \Omega_{j,2} \end{cases}
\end{gathered} \tag{3.4}$$

$$\begin{gathered}
\begin{cases} u_i = u_{j,1}, & \text{on } \Sigma_{i,j} \\ \frac{\partial u_i}{\partial n_i} = -\frac{\partial u_{j,1}}{\partial n_{j,1}}, & \text{on } \Sigma_{i,j} \end{cases} \\
\begin{cases} u_{j,1} = u_{j,2}, & \text{on } \Sigma_j \\ \frac{\partial u_{j,1}}{\partial n_{j,1}} = -\frac{\partial u_{j,2}}{\partial n_{j,2}}, & \text{on } \Sigma_j. \end{cases}
\end{gathered} \tag{3.5}$$

Although (3.4) is restricted to two building blocks, it provides a sense of how the equation can be broken down across a variety of subdomains. As such, to avoid unnecessarily complicated and repeated equations, the formulations for the rest of this chapter are assumed to be abstracted from (3.4) according to their defined domains.

3.2 Method of Difference Potentials for Rod Subdomains

In this section, we introduce the method of difference potentials in the context of the rod subdomain. The individual pieces of the formulation have been rigorously introduced and explored in previous works [Albright et al., 2017; D. S. Britt et al., 2013; Medvinsky et al., 2012, 2013, 2016; Reznik, 1982; Ryaben' kii, 2002]. The case of the empty subdomain is covered in Section 2.1.1 (see also [North et al., 2022]). In this section we present these individual pieces within a unified framework as they relate to the decomposition of the PCRR. The following sections are intended to only provide the definitions relevant to the topic of the rod subdomain. Many of the equations and definitions are direct extensions of the work in Section 2.1 and [North et al., 2022], and are presented here without discussion or justification. Refer to [North et al., 2022] and the citations therewithin for a more complete presentation of the material.

3.2.1 Auxiliary Problems

Computing the Calderon's potentials and projections requires setting an appropriate auxiliary problem (AP), which serves to define an inverse to the discrete Helmholtz operator. The AP is formulated on an auxiliary domain that contains the physical domain as a subset. It should be uniquely solvable and well-posed, and should admit an efficient numerical solution. Otherwise, the AP can be arbitrary [Ryaben' kii, 2002] and our specific choice is made for reasons of convenience. We start by briefly introducing the auxiliary problem for the empty subdomain (as in Section 2.1.1.1 and [North et al., 2022]) to show how the same AP can be used for the part of the rod subdomain that is exterior to the circle ($\Omega_{j,1}$ in Figure 3.1). Additionally, we introduce an AP for the interior of the circle in the rod subdomain.

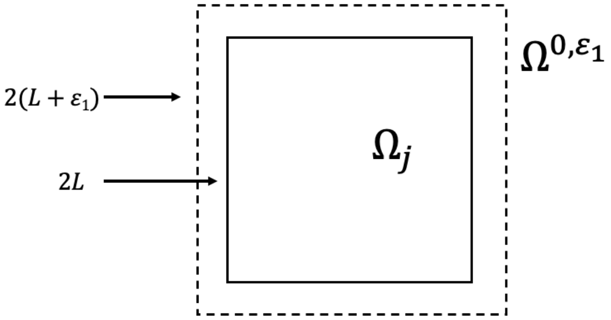


Figure 3.3 Auxiliary domain Ω^0 surrounding an empty subdomain Ω_j.

Let Ω_j be a square with side length $2L$, centered about the origin. Define the auxiliary domain $\Omega^{0,\varepsilon_1} \supset \Omega_j$ to be the square of side length $2(L+\varepsilon_1)$ for some small $\varepsilon_1 > 0$ (Figure 3.3, practical choices for ϵ_1 are discussed in Section 3.3). Consider the following auxiliary problem:

$$\begin{cases} \mathscr{L}_j u \equiv (\Delta + k_j^2) u = g & (x,y) \in \Omega^{0,\varepsilon_1}, \\ u = 0 & y = \pm(L+\varepsilon_1), \\ \frac{\partial u}{\partial x} + i k_j u = 0 & x = L+\varepsilon_1, \\ \frac{\partial u}{\partial x} - i k_j u = 0 & x = -L-\varepsilon_1. \end{cases} \tag{3.6}$$

The AP (3.6) is defined for any right-hand side (RHS) g. The Sommerfeld-like conditions on the

$x-$bounds are chosen to ensure the AP is well-posed by shifting the spectrum of the subproblem to the complex plane, ensuring resonance is avoided for any real wavenumber k_j. The Dirichlet conditions on the $y-$bounds permit an efficient numerical solution by means of a sine-FFT in the $y-$direction paired with a tridiagonal solver.

Let Ω_j be a rod subdomain (Figure 3.1) such that the boundary $\partial\Omega_j$ is again a square with side length $2L$ and is centered about the origin, and let the circle inside the rod subdomain $\Omega_{j,2}$ have radius R. Further, let the material discontinuity be characterized by the following definition of the wavenumber k_j:

$$k_j = \begin{cases} k_{j,1} & (x,y) \in \Omega_{j,1}, \\ k_{j,2} & (x,y) \in \Omega_{j,2}. \end{cases}$$

The rod subdomain requires the definition of two auxiliary problems. The first of these APs is equivalent to (3.6) and accounts for the solution inside of $\Omega_{j,1}$. Note that even though k_j has a discontinuity, this AP still uses the constant value assigned to $\Omega_{j,1}$, i.e.

$$\begin{cases} \mathscr{L}_{j,1} u \equiv \Delta u + k_{j,1}^2 u = g & (x,y) \in \Omega^{0,\varepsilon_1}, \\ u = 0 & y = \pm(L+\varepsilon_1), \\ \frac{\partial u}{\partial x} + i k_{j,1} u = 0 & x = L + \varepsilon_1, \\ \frac{\partial u}{\partial x} - i k_{j,1} u = 0 & x = -L - \varepsilon_1. \end{cases} \tag{3.7a}$$

The second AP handles the interior of the circle and requires its own auxiliary domain. Let $\varepsilon_2 > 0$ be given and define Ω^{0,ε_2} to be a square of side length $2(R+\varepsilon_2)$ so that $\Omega^{0,\varepsilon_2} \supset \Omega_{j,2}$. Consider the following AP:

$$\begin{cases} \mathscr{L}_{j,2} u \equiv \Delta u + k_{j,2}^2 u = g & (x,y) \in \Omega^{0,\varepsilon_2}, \\ u = 0 & y = \pm(R+\varepsilon_2), \\ \frac{\partial u}{\partial x} + i k_{j,2} u = 0 & x = R + \varepsilon_2, \\ \frac{\partial u}{\partial x} - i k_{j,2} u = 0 & x = -R - \varepsilon_2. \end{cases} \tag{3.7b}$$

Figure 3.4 depicts both auxiliary domains, Ω^{0,ε_1} and Ω^{0,ε_2} in relation to their respective subdomains. The Helmholtz operators $\mathscr{L}_{j,1} = (\Delta + k_{j,1}^2)$ and $\mathscr{L}_{j,2} = (\Delta + k_{j,2}^2)$ of the rod subdomain APs (3.7a) and (3.7b) are discretized using the left-hand side of the finite difference scheme (1.9) with the discrete counterparts denoted by $\mathscr{L}_{j,1}^{(h)}$ and $\mathscr{L}_{j,2}^{(h)}$, respectively. In order to preserve the overall accuracy of the scheme, the boundary conditions from the APs also need to be discretized to fourth-order accuracy. The Dirichlet conditions set at the $y-$bounds, $y = \pm(L+\varepsilon_1)$ and $\pm(R+\varepsilon_2)$, are trivial as the boundary nodes simply are set to zero, i.e., for a discretization with $M+1$ nodes in the $x-$direction and $N+1$ nodes in the $y-$direction, set

$$u_{m,0} = u_{m,N} = 0, \qquad m = 0, \ldots, M. \tag{3.8a}$$

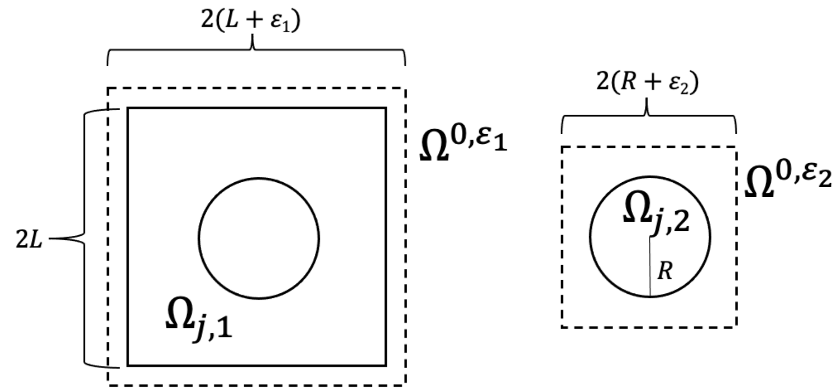


Figure 3.4 Auxiliary domains Ω^{0,ε_1} and Ω^{0,ε_2} depicted containing their respective domains. Ω_j can be an empty or rod subdomain.

For the conditions on the x—bounds, we use the following discretization, derived for the variable coefficient Helmholtz equation in [S. Britt et al., 2011] and simplified for the constant-coefficient case in [D. S. Britt et al., 2013]:

$$\begin{aligned}&\Bigg(\frac{u_{M,n}-u_{M-1,n}}{h}-\frac{1}{6h}\big(u_{M,n+1}-u_{M-1,n+1}+u_{M,n-1}-u_{M-1,n-1}\\&-2\big(u_{M,n}-u_{M-1,n}\big)\big)-\frac{k^2h}{24}\big(u_{M,n}-u_{M-1,n}\big)\Bigg)\\&+ik\Bigg(\frac{u_{M,n}-u_{M-1,n}}{h}+\frac{h^2k^2}{8}u_{M-\frac{1}{2},n}\\&+\frac{u_{M-\frac{1}{2},n+1}-2u_{M-\frac{1}{2},n}+u_{M-\frac{1}{2},n-1}}{2}\Bigg)=0\end{aligned}\tag{3.8b}$$

$$\begin{aligned}&\Bigg(\frac{u_{1,n}-u_{0,n}}{h}-\frac{1}{6h}\big(u_{1,n+1}-u_{0,n+1}+u_{1,n-1}-u_{0,n-1}-2\big(u_{1,n}-u_{0,n}\big)\big)\\&-\frac{k^2h}{24}\big(u_{1,n}-u_{0,n}\big)\Bigg)\\&-ik\Bigg(\frac{u_{1,n}-u_{0,n}}{h}+\frac{h^2k^2}{8}u_{\frac{1}{2},n}+\frac{u_{\frac{1}{2},n+1}-2u_{\frac{1}{2},n}+u_{\frac{1}{2},n-1}}{2}\Bigg)=0.\end{aligned}\tag{3.8c}$$

The discrete APs for the rod subdomain can now be expressed as $\mathscr{L}_{j,1}^{(h)}u^{(h)}=g^{(h)}$ and $\mathscr{L}_{j,2}^{(h)}u^{(h)}=g^{(h)}$, subject to the discrete boundary conditions (3.8). Since (3.7a) and (3.6) are basically the same, $\mathscr{L}_{j,1}^{(h)}u^{(h)}=g^{(h)}$ also discretizes (3.6). Like their continuous counterparts (3.7) (and (3.6)), the discrete APs have unique solutions for any right-hand side $g^{(h)}$. Indeed, each of these APs can be solved directly by a combination of a FFT in the y direction and a tridiagonal elimination in the x direction. The resulting solutions define the inverse operators $\mathscr{G}_{j,l}^{(h)}$: $u^{(h)}=\mathscr{G}_{j,l}^{(h)}g^{(h)}$ for $l=1,2$. By design, the operators $\mathscr{G}_{j,l}^{(h)}$ apply to an arbitrary $g^{(h)}$. If f represents a physical source function, see equation (3.1a), then

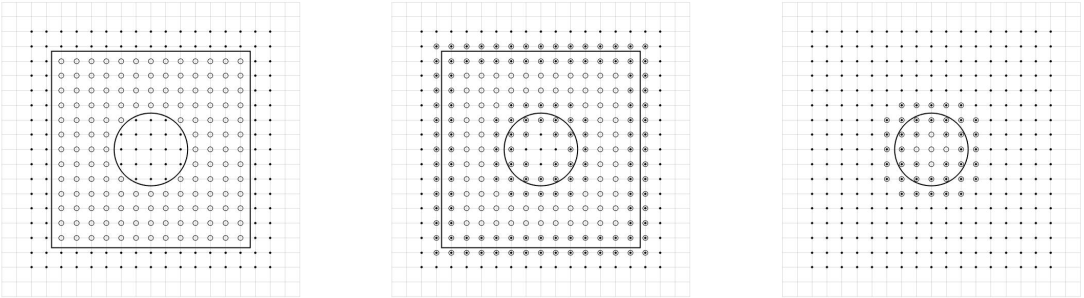
$$g^{(h)}=\mathscr{B}^{(h)}f\overset{\text{def}}{=}f_{m,n}+\frac{1}{12}\big(f_{m+1,n}+f_{m-1,n}+f_{m,n+1}+f_{m,n-1}-4f_{m,n}\big)\tag{3.9}$$

where $\mathscr{B}^{(h)}$ denotes the application of the right-hand side stencil from (1.9). However, for constructing the difference potentials the operators $\mathscr{G}^{(h)}_{j,l}$ will need to be applied to right-hand sides $g^{(h)}$ besides (3.9), see Sections 3.2.2 and 3.2.3.

3.2.2 Difference Potentials and Operators

In this section, we define the grid sets that serve as discrete analogues for the various parts of the rod subdomain. These grid sets are used to define the difference potential, which is the key operator and discrete analogue to the Calderon's operator. The difference potential in turn defines the boundary projection operator. Following the example of [D. S. Britt et al., 2013; Medvinsky et al., 2012; North et al., 2022], we also present the equation-based extension operator that links the continuous data to the discrete nodes as well as the spectral representation of the solution at the boundary. Throughout this section, we only consider a single rod subdomain Ω_j. As such, the subscript indices are simplified and restricted to $l \in \{1,2\}$ to indicate whether the grid set or operator is constructed for $\Omega_{j,1}$ or $\Omega_{j,2}$, dropping the first index j altogether.

Let $\mathbb{N}^0_l$ be a uniformly spaced Cartesian grid over Ω^{0,ε_l} and $\mathbb{M}^0_l \subset \mathbb{N}^0_l$ be the set of nodes that are strictly inside of $\mathbb{N}^0_l$ (i.e. excluding the outermost layer). The nodes of $\mathbb{M}^0_l$ that are inside of the physical domain Ω_l are $\mathbb{M}^+_l = \mathbb{M}^0_l \cap \Omega_l$, and the exterior nodes are $\mathbb{M}^-_l = \mathbb{M}^0_l \cap \Omega^{\complement}_l$ (the complement of Ω_l). Note that specifically, the nodes of $\mathbb{M}^+_1$ lie inside the boundaries of the square and outside the circle (Figure 3.5a), while the nodes of $\mathbb{M}^+_2$ fall inside the circle itself. Let $\mathbb{N}^+_l$ ($\mathbb{N}^-_l$) consist of the nodes obtained when applying $\mathscr{L}^{(h)}_l$ to the nodes of $\mathbb{M}^+_l$ ($\mathbb{M}^-_l$). The sets $\mathbb{N}^+_l$ and $\mathbb{N}^-_l$ are depicted in Figures 3.5b and 3.5c. Then $\mathbb{N}^+_l$ and $\mathbb{N}^-_l$ will have a set of overlapping nodes near the boundary of Ω_l. This set of nodes is $\gamma_l = \mathbb{N}^+_l \cap \mathbb{N}^-_l$, and is referred to as the discrete boundary.



(a) ○—$\mathbb{M}^+_1$ ●—$\mathbb{M}^-_1$ **(b)** ○—$\mathbb{N}^+_1$ ●—$\mathbb{N}^-_1$ ⊙—γ_1 **(c)** ○—$\mathbb{N}^+_2$ ●—$\mathbb{N}^-_2$ ⊙—γ_2

Figure 3.5 Depictions of several grids sets. For each, $\mathbb{N}^0$ is the underlying mesh of grid nodes.

To simplify notation, the index l will also be dropped. Many of the definitions and results in Section 3.2.2 are given identically for both parts of the subdomain, so grid sets and operators should be interpreted as consistent within their particular subdomain.

3.2.2.1 Difference Potentials and Projections

Consider a grid function ξ_γ specified on the discrete boundary γ. We then define the *difference potential with density* ξ_γ as

$$P_{\mathbb{N}^+}\xi_\gamma \overset{\text{def}}{=} w - \mathscr{G}^{(h)}\big(\mathscr{L}^{(h)} w\big|_{\mathbb{M}^+}\big), \quad \text{where} \quad w = \begin{cases} \xi_\gamma & \text{on } \gamma, \\ 0 & \text{on } \mathbb{N}^0\backslash\gamma. \end{cases} \tag{3.10}$$

The difference potential (3.10) truncated to the grid boundary γ defines the *discrete boundary projection operator* P_γ

$$P_\gamma \xi_\gamma \overset{\text{def}}{=} \big(P_{\mathbb{N}^+}\xi_\gamma\big)\Big|_\gamma \equiv Tr^{(h)}(P_{\mathbb{N}^+}\xi_\gamma) \tag{3.11}$$

where the operator $Tr^{(h)}$ refers to trace (truncation) of the grid set to its associated boundary γ. The projection P_γ given by (3.11) has the following key property:

Theorem 1 (Ryaben'kii) *A grid function ξ_γ satisfies the difference boundary equation with projection (BEP)*

$$P_\gamma \xi_\gamma + Tr^{(h)}\mathscr{G}^{(h)} g = \xi_\gamma \tag{3.12}$$

if and only if there is a solution u on $\mathbb{N}^+$ to the finite difference equation $\mathscr{L}^{(h)} u = g$ such that ξ_γ is the trace of u on the grid boundary γ, i.e. $\xi_\gamma = Tr^{(h)} u$.

The proof of Theorem 1 (as well as that of the projection property, $P_\gamma^2 = P_\gamma$) can be found in [Ryaben' kii, 2002]. If the BEP (3.12) holds for ξ_γ on γ, then the solution u can be reconstructed on $\mathbb{N}^+$ by means of the discrete generalized Green's formula:

$$u = P_{\mathbb{N}^+}\xi_\gamma + \mathscr{G}^{(h)} g. \tag{3.13}$$

3.2.2.2 Equation-Based Extension

To guarantee the grid function ξ_γ in (3.13) accurately represents the continuous boundary data of the solution, we use an *extension operator*. For a given boundary Γ (either $\Gamma = \partial\Omega_1$ or $\Gamma = \partial\Omega_2$), consider a pair of functions (ξ_0, ξ_1) defined along Γ. One can consider these two functions as the Dirichlet and Neumann data, respectively, of some underlying function $v = v(x, y)$ on Ω^0:

$$(\xi_0, \xi_1) = \left(v, \frac{\partial v}{\partial \mathbf{n}}\right)\Bigg|_\Gamma .$$

Near Γ, the function v can be expanded into a truncated Taylor series, where ρ represents the (unsigned) distance from the point of evaluation to the nearest point on Γ:

$$v(x, y) \overset{\text{def}}{=} v|_\Gamma + \rho \left.\frac{\partial v}{\partial \mathbf{n}}\right|_\Gamma + \frac{\rho^2}{2}\left.\frac{\partial^2 v}{\partial \mathbf{n}^2}\right|_\Gamma + \frac{\rho^3}{6}\left.\frac{\partial^3 v}{\partial \mathbf{n}^3}\right|_\Gamma + \frac{\rho^4}{24}\left.\frac{\partial^4 v}{\partial \mathbf{n}^4}\right|_\Gamma . \tag{3.14}$$

Discussion and results pertaining to the number of terms included in this expansion can be found in [D. S. Britt et al., 2013; Medvinsky et al., 2012; North et al., 2022], and trace back to formal results by Reznik [Reznik, 1982, 1983].

When the functions ξ_0 and ξ_1 are given, the first two terms of (3.14) can be computed directly. For the higher-order derivative terms, formulas can be produced by using equation-based differentiation applied to the Helmholtz equation (3.1a), where we assume v is a solution and v and $\frac{\partial v}{\partial \mathbf{n}}$ are known analytically on Γ.

For example, when Γ is the right side of a square (where $x = X$ is constant), the outward normal derivative coincides with the positive $x-$derivative. By substituting $v = \xi_0$ and $\frac{\partial v}{\partial \mathbf{n}} = \xi_1$, we can obtain the following sequence of formulas (see Section 2.1.1.3 or [North et al., 2022] for details on this derivation):

$$\begin{aligned}
v(X,y) &= \xi_0(y),\\
\frac{\partial v}{\partial x}(X,y) &= \xi_1(y),\\
\frac{\partial^2 v}{\partial x^2}(X,y) &= f(X,y) - \xi_0''(y) - k^2\xi_0(y),\\
\frac{\partial^3 v}{\partial x^3}(X,y) &= \frac{\partial f}{\partial x}(X,y) - \xi_1''(y) - k^2\xi_1(y),\\
\frac{\partial^4 v}{\partial x^4}(X,y) &= \frac{\partial^2 f}{\partial x^2}(X,y) - \frac{\partial^2 f}{\partial y^2}(X,y) - k^2 f(X,y) + \xi_0^{(4)}(y) + 2k^2\xi_0^{(2)}(y) + k^4\xi_0(y).
\end{aligned} \tag{3.15}$$

Such formulas can also be obtained for the other sides of a square by substituting the boundary coordinates and appropriately identifying the outward normal derivatives, for example on the left side the outward normal derivative coincides with the *negative* $x-$derivative.

Additionally, when considering the rod subdomain there is a circular piece of the boundary that requires its own extension. Using the same general expansion (3.14), we recast the Helmholtz equation (3.1a) in polar coordinates (r, θ):

$$\frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} = f.$$

Consider the case of the interior of the rod so that the outward normal derivative coincides with the positive $r-$derivative (at $r = R$ a constant). The functions ξ_0 and ξ_1 then depend on θ and the following equations are obtained (as in [D. S. Britt et al., 2013]):

$$\begin{aligned}
v(R,\theta) &= \xi_0(\theta), \\
\frac{\partial v}{\partial r}(R,\theta) &= \xi_1(\theta), \\
\frac{\partial^2 v}{\partial r^2}(R,\theta) &= f(R,\theta) - \frac{1}{R}\xi_1(\theta) - \frac{1}{R^2}\xi_0''(\theta) - k^2\xi_0(\theta), \\
\frac{\partial^3 v}{\partial r^3}(R,\theta) &= \frac{\partial f}{\partial r}(R,\theta) - \frac{1}{R}\frac{\partial^2 v}{\partial r^2}(R,\theta) + \frac{1}{R^2}\xi_1(\theta) + \frac{2}{R^3}\xi_0''(\theta) - \frac{1}{R^2}\xi_1''(\theta) - k^2\xi_1, \\
\frac{\partial^4 v}{\partial r^4}(R,\theta) &= \frac{\partial^2 f}{\partial r^2} - \frac{2}{R^3}\xi_1(\theta) + \left(\frac{2}{R^2} - k^2\right)\frac{\partial^2 v}{\partial r^2}(R,\theta) - \frac{1}{R}\frac{\partial^3 v}{\partial r^3}(R,\theta) \\
&\quad + \left(\frac{k^2}{R^2} - \frac{6}{R^4}\right)\xi_0''(\theta) + \frac{5}{R^3}\xi_1''(\theta) - \frac{1}{R^2}\left(\frac{\partial^2 f}{\partial \theta^2}(R,\theta) - \frac{1}{R^2}\xi_0^{(4)}(\theta)\right),
\end{aligned} \tag{3.16}$$

where $\xi_0^{(4)}(\theta)$ represents the fourth derivative of $\xi_0(\theta)$ with respect to θ. Note that when considering the circle from outside the rod, the outward normal derivative will instead coincide with the *negative* r—derivative.

The expressions from either (3.15), (3.16), or their appropriate analogues can be substituted into (3.14) to construct $v = v(x, y)$ from any pair of functions (ξ_0, ξ_1) defined on Γ. Sampling this function v only at nodes of the grid boundary γ defines the extension operator **Ex** that gives the grid function ξ_γ:

$$\xi_\gamma = \mathbf{Ex}(\xi_0, \xi_1) = v|_\gamma.$$

As seen in (3.15) and (3.16) the extension operator depends on the source term f. The constribution from the source term makes **Ex** an affine operator:

$$\mathbf{Ex}\xi_\Gamma = \mathbf{Ex}^{(H)}(\xi_0, \xi_1) + \mathbf{Ex}^{(I)} f.$$

The $\mathbf{Ex}^{(H)}$ (homogeneous) operator acts only on the input functions (ξ_0, ξ_1), while the $\mathbf{Ex}^{(I)}$ (inhomogneous) operator is dependent on the source function f from the original problem (3.1a).

We emphasize that although (3.1a) was used to construct the expressions of (3.15) and (3.16), $\xi_\Gamma = (\xi_0, \xi_1)$ does not need to represent the Cauchy data of a solution u to (3.1a) in order to apply the operator **Ex**. However, if ξ_Γ does correspond to the trace of a solution u, i.e. $\xi_\Gamma = \left(u, \frac{\partial u}{\partial \mathbf{n}}\right)\big|_\Gamma$, then $\xi_\gamma = \mathbf{Ex}\xi_\Gamma$ approximates u with fifth-order accuracy near Γ with respect to the grid size, specifically at the nodes of ξ_γ.

3.2.2.3 Spectral Representation at the Boundary

Following the example Section 2.1.1.4 we define a series representation of the boundary data, with the additional consideration of the central circle. Consider the breakdown of the boundary Γ in Figure 3.6. For each Γ_i, choose a set of M_i basis functions $\{\psi_j\}_{j=1}^{M_i}$ such that each ψ_j is supported only on its corresponding Γ_i.

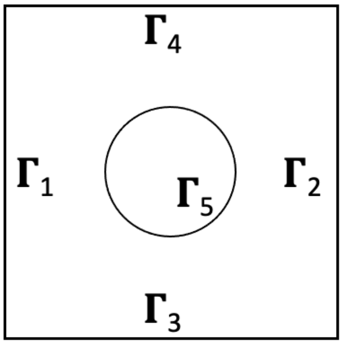


Figure 3.6 Breakdown of the portions of Γ in the rod subdomain.

Generally speaking, each set $\{\psi_j\}_{j=1}^{M_i}$ can be chosen independently. For this particular setup, we use Chebyshev polynomials along the square edges $\Gamma_1,\ldots,\Gamma_4$ and Fourier basis functions around the central circle boundary Γ_5. To further simplify equations later on, we take each Chebyshev basis to have the same dimension, M_c, and pick the Fourier basis along Γ_5 such that $M_5 = |\{\psi_j\}_{-M_r}^{M_r}| = 2M_r+1$. Given these specifications, we can define a larger set of basis functions $\{\Psi_j\}$ such that

$$\{\Psi_j\}_{j=1}^{M} = \underbrace{\{\psi_j\}_{j=1}^{M_c} \cup \{\psi_j\}_{j=1}^{M_c} \cup \{\psi_j\}_{j=1}^{M_c} \cup \{\psi_j\}_{j=1}^{M_c}}_{\Gamma_1,\ldots,\Gamma_4} \cup \underbrace{\{\psi_j\}_{-M_r}^{M_r}}_{\Gamma_5} \tag{3.17}$$

where $M = 4M_c + 2M_r + 1$. Using the basis functions $\{\Psi_j\}$ defined in (3.17), consider the following pairs:

$$\boldsymbol{\Psi}_j^{(0)} = (\Psi_j, 0), \quad \boldsymbol{\Psi}_j^{(1)} = (0, \Psi_j), \quad j = 1,\ldots,M. \tag{3.18}$$

For boundary data of the form $\xi_\Gamma = (\xi_0, \xi_1)$, the pairs defined in (3.18) allow the components ξ_0 and ξ_1 to be expanded individually. Note that the set of functions used to define $\boldsymbol{\Psi}_j^{(0)}$ could be different from those used to define $\boldsymbol{\Psi}_j^{(1)}$, but for simplicity we use the same sets. Assume the boundary data of the solution we seek is given in the form $\xi_\Gamma = \left(u, \frac{\partial u}{\partial \mathbf{n}}\right)\big|_\Gamma$, then that data can be expressed as the following series representation

$$\xi_\Gamma = \sum_{j=1}^{M} \mathbf{c}_j^{(0)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \boldsymbol{\Psi}_j^{(1)} \tag{3.19}$$

where $\mathbf{c}^{(0)}$ and $\mathbf{c}^{(1)}$ are the unknown expansion coefficients of the Dirichlet and Neumann data, respectively. For smooth boundary data, the Chebyshev and Fourier expansions converge rapidly, making the truncated terms of the expansion negligible with respect to the accuracy attainable on the grid for relatively small values of M_c and M_r. Specific values of M_c and M_r used in our simulations are presented in Appendix A.

The series (3.19) is defined in relation to the subdomain $\Omega_{j,1}$ (as in Figure 3.6). One could similarly define a series representation of the solution at the boundary of $\Omega_{j,2}$ as well, but instead we remark that along $\overline{\Omega}_{j,1} \cap \overline{\Omega}_{j,2}$ (i.e. the circle) we know that $u_{j,1} = u_{j,2}$ and $\frac{\partial u_{j,1}}{\partial \mathbf{n_{j,1}}} = -\frac{\partial u_{j,2}}{\partial \mathbf{n_{j,2}}}$, allowing the portion (3.17) corresponding to Γ_5 to be reused when considering $\Omega_{j,2}$. This notion is utilized and further explained in Section 3.2.3.2.

3.2.3 Linear System

Consider a rod subdomain $\Omega_j = \Omega_{j,1} \cup \Omega_{j,2}$ with $\Gamma = \partial \Omega_{j,1}$. Let u be a solution to (3.1) over Ω_j and let $\xi_\Gamma = \left(u, \frac{\partial u}{\partial \mathbf{n}}\right)\big|_\Gamma$ be the trace of the Dirichlet and Neumann data of the solution at the boundary. Let ξ_γ be given by applying the **Ex** operator to the series representation (3.19) of the solution at the boundary of $\Omega_{j,1}$:

$$\begin{aligned} \xi_\gamma := \mathbf{Ex}\,\xi_\Gamma &= \mathbf{Ex}^{(H)}\left(\sum_{j=1}^{M} \mathbf{c}_j^{(0)} \mathbf{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \mathbf{\Psi}_j^{(1)}\right) + \mathbf{Ex}^{(I)} f \\ &= \sum_{j=1}^{M} \mathbf{c}_j^{(0)} \mathbf{Ex}^{(H)} \mathbf{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} \mathbf{Ex}^{(H)} \mathbf{\Psi}_j^{(1)} + \mathbf{Ex}^{(I)} f. \end{aligned} \tag{3.20}$$

The right-hand side of (3.20) can then be substituted into the discrete BEP (3.12) for $\Omega_{j,1}$. After rearranging terms, the following system is obtained:

$$\begin{aligned} &\sum_{j=1}^{M} \mathbf{c}_j^{(0)} (P_\gamma - I_\gamma) \mathbf{Ex}^{(H)} \mathbf{\Psi}_j^{(0)} + \sum_{j=1}^{M} \mathbf{c}_j^{(1)} (P_\gamma - I_\gamma) \mathbf{Ex}^{(H)} \mathbf{\Psi}_j^{(1)} \\ &\qquad\qquad = \underbrace{(I_\gamma - P_\gamma) \mathbf{Ex}^{(I)} f - Tr^{(h)} \mathcal{G}^{(h)} \mathcal{B}^{(h)} f}_{F} \end{aligned} \tag{3.21}$$

where the coefficients $\mathbf{c}_j^{(0)}$ and $\mathbf{c}_j^{(1)}$ are the unknowns and I_γ represents the identity operator on γ. Letting F represent the right-hand side of (3.21) allows this equation to be expressed more compactly as

$$Q\mathbf{c} = F \tag{3.22}$$

where Q and c have the following forms:

$$\begin{aligned} Q = \Big[&\underbrace{(P_\gamma - I_\gamma) \mathbf{Ex} \mathbf{\Psi}_1^{(0)}, \ldots, (P_\gamma - I_\gamma) \mathbf{Ex} \mathbf{\Psi}_M^{(0)},}_{Q^{(0)}} \\ &\underbrace{(P_\gamma - I_\gamma) \mathbf{Ex} \mathbf{\Psi}_1^{(1)}, \ldots, (P_\gamma - I_\gamma) \mathbf{Ex} \mathbf{\Psi}_M^{(1)}}_{Q^{(1)}} \Big], \\ \mathbf{c} = &[\underbrace{\mathbf{c}_1, \ldots, \mathbf{c}_M}_{c^{(0)\mathsf{T}}}, \underbrace{\mathbf{c}_{M+1}, \ldots, \mathbf{c}_{2M}}_{c^{(1)\mathsf{T}}}]^{\mathsf{T}} \end{aligned} \tag{3.23}$$

where the dimensions of Q are $|\gamma| \times 2M$. In addition to (3.22), consider the contributions from the rod, $\Omega_{j,2}$. Let $\Gamma_r = \partial \Omega_{j,2}$ and repeat the steps above for the boundary data ξ_{Γ_r} to obtain a separate system corresponding to the scattering rod:

$$Q_r \mathbf{c}_r = F_r. \tag{3.24}$$

In (3.24), $\mathbf{c}_r$ represents the coefficients of the expansion of the boundary data along the circle Γ_r and F_r is an analogue to the right-hand side of (3.21) when the discrete boundary is γ_r instead of γ. The matrix $Q_r = \left[Q_r^{(0)}, Q_r^{(1)}\right]$ has a form similar to Q from (3.23), with dimensions $|\gamma_r| \times (4M_r + 2)$. The expansions of the Dirichlet and Neumann data along Γ_r each use $2M_r + 1$ Fourier basis functions since Γ_r contains only the circular boundary of the rod.

Note that the solutions of (3.22) and (3.24) are not unique because the BEPs (3.12) were derived only from the Helmholtz equation (3.1a). Additional equations are needed to supplement the system and make the solution $\mathbf{c}$ unique, although the exact nature of these conditions depends on the context. Section 3.2.3.1 explains how to derive such equations from local ABCs given on the exterior edges of $\partial\Omega_{j,1}$. When an edge of $\partial\Omega_{j,1}$ creates an interface with another subdomain, conditions can be derived as in Section 2.1.2.2. In Section 3.2.3.2, we explain how to use the interface conditions between $\Omega_{j,1}$ and the rod $\Omega_{j,2}$.

3.2.3.1 Resolving the Local ABCs

Consider the second-order Engquist-Majda ABC (1.8):

$$ik\frac{\partial u}{\partial \mathbf{n}} - k^2 u - \frac{1}{2}\frac{\partial^2 u}{\partial \tau^2} = 0. \tag{3.25}$$

Our goal is to substitute the series representations of the Dirichlet and Neumann data into (3.25). Without loss of generality, consider the series representation of the boundary data along one side of the exterior of the square subdomain and pass it into condition (3.25):

$$\begin{aligned} ik\left(\sum_{l=1}^{M_c} \mathbf{c}_l^{(1)}\psi_l\right) - k^2\left(\sum_{l=1}^{M_c} \mathbf{c}_l^{(0)}\psi_l\right) - \frac{1}{2}\frac{\partial^2}{\partial\tau^2}\left(\sum_{l=1}^{M_c} \mathbf{c}_l^{(0)}\psi_l\right) &= 0 \\ \sum_{l=1}^{M_c}\left(ik\mathbf{c}_l^{(1)} - k^2\mathbf{c}_l^{(0)}\right)\psi_l - \frac{1}{2}\left(\sum_{l=1}^{M_c} \mathbf{c}_l^{(0)}\frac{\partial^2}{\partial\tau^2}\psi_l\right) &= 0. \end{aligned} \tag{3.26}$$

The formulation does not include a direct substitution for tangential derivatives (only the solution and its normal derivative). Instead, the second tangential derivative $\frac{\partial^2}{\partial\tau^2}$ is passed onto the basis functions and computed directly.

Along the edge of the square, tangential derivatives of the Chebyshev basis functions $\{\psi_l\}_1^{M_c}$ correspond to standard derivatives with respect to the argument. Given the Chebyshev coefficients of a function (i.e. $\{\mathbf{c}_l^{(0)}\}$), the coefficients of the second derivative (i.e. $\{d_l\}$) can be computed with the following formula, adapted from [Karageorghis, 1988]:

$$a_l d_l = 2\cdot\sum_{j=1}^{\infty} j\cdot(l+j)\cdot(l+2j)\cdot\mathbf{c}_{l+2j}^{(0)},\quad a_l = \begin{cases} 2, & l=0 \\ 1, & l>0. \end{cases}$$

The needed coefficients, $\{\mathbf{c}_l^{(0)}\}$ are unknown, so the formula is instead applied to each basis function individually to construct a matrix $A = \left[A_{i,j}\right]$ that represents the second derivative of the sum:

$$\begin{aligned}
\frac{\partial^2}{\partial\tau^2}\sum_{l=1}^{M_c}\mathbf{c}_l^{(0)}\psi_l &= \sum_{l=1}^{M_c}\mathbf{c}_l^{(0)}\frac{\partial^2}{\partial\tau^2}\psi_l \\
&= \sum_{l=1}^{M_c}\mathbf{c}_l^{(0)}\left(\sum_{j=1}^{M_c}A_{l,j}\psi_j\right) \\
&= \sum_{l=1}^{M_c}\sum_{j=1}^{M_c}\mathbf{c}_l^{(0)}A_{l,j}\psi_j \\
&= \sum_{j=1}^{M_c}\psi_j\left(\sum_{l=1}^{M_c}\mathbf{c}_l^{(0)}A_{l,j}\right) \\
&= \sum_{j=1}^{M_c}\left(A_{*,j}\right)^{\mathsf{T}}\mathbf{c}^{(0)}\psi_j
\end{aligned}\tag{3.27}$$

where row l of A corresponds to the coefficients of the second derivative of ψ_l, and $A_{*,j}$ indicates column j of A. Substituting (3.27) into (3.26) yields:

$$\begin{aligned}
\sum_{l=1}^{M_c}\left(ik\mathbf{c}_l^{(1)} - k^2\mathbf{c}_l^{(0)}\right)\psi_l - \frac{1}{2}\left(\sum_{l=1}^{M_c}\left(A_{*,l}\right)^{\mathsf{T}}\mathbf{c}^{(0)}\psi_l\right) &= 0 \\
\sum_{l=1}^{M_c}\left(ik\mathbf{c}_l^{(1)} - k^2\mathbf{c}_l^{(0)} - \frac{1}{2}\left(A_{*,l}\right)^{\mathsf{T}}\mathbf{c}^{(0)}\right)\psi_l &= 0.
\end{aligned}$$

Orthogonality of the Chebyshev basis tells us that for each of $l = 1,\ldots,M_c$, we get a condition that relates the coefficients:

$$ik\mathbf{c}_l^{(1)} - k^2\mathbf{c}_l^{(0)} - \frac{1}{2}\left(A_{*,l}\right)^{\mathsf{T}}\mathbf{c}^{(0)} = 0. \tag{3.28}$$

The conditions of (3.28) are coupled due to the third term, preventing them from being handled individually. Instead, consider them together as a matrix equation

$$\begin{aligned}
ik\mathbf{c}^{(1)} - k^2\mathbf{c}^{(0)} - \frac{1}{2}A^{\mathsf{T}}\mathbf{c}^{(0)} &= \vec{0} \\
-\left(k^2 I + \frac{1}{2}A^{\mathsf{T}}\right)\mathbf{c}^{(0)} + ik\mathbf{c}^{(1)} &= \vec{0}.
\end{aligned}\tag{3.29}$$

The relationship between $\mathbf{c}^{(0)}$ and $\mathbf{c}^{(1)}$ in (3.29) provides M_c supplemental conditions for system

(3.22). This process is repeated for each side of the subdomain that contains an appropriate ABC.

It is well-known that matrices such as A in (3.29) can lead to poor conditioning, particularly when used in spectral solvers [Hesthaven et al., 2007]. This problem more readily manifests itself in the presence of higher-order tangential derivatives (such as those that appear in higher-order Engquist-Majda ABCs). The numerical results we present in Section 3.3.1 suggest that such poor conditioning is avoided, for practical purposes, in the second-order condition we have chosen.

3.2.3.2 Resolving Interface Conditions Along the Rod

Let $\Gamma_5 \subset \partial\Omega_{j,1}$ be the circle from the boundary of $\Omega_{j,1}$ (as in Figure 3.6) and $\Gamma_r = \partial\Omega_{j,2}$. Consider the following expansions of the boundary data along Γ_5 and Γ_r, respectively:

$$\begin{aligned} \xi_{\Gamma_5} &= \sum_{l=-M_r}^{M_r} \mathbf{c}_{5,l}^{(0)} \boldsymbol{\psi}_l^{(0)} + \sum_{l=-M_r}^{M_r} \mathbf{c}_{5,l}^{(1)} \boldsymbol{\psi}_l^{(1)}, \\ \xi_{\Gamma_r} &= \sum_{l=-M_r}^{M_r} \mathbf{c}_{r,l}^{(0)} \boldsymbol{\psi}_l^{(0)} + \sum_{l=-M_r}^{M_r} \mathbf{c}_{r,l}^{(1)} \boldsymbol{\psi}_l^{(1)}. \end{aligned} \tag{3.30}$$

Note that $\boldsymbol{\psi}_l^{(0)} = \left(\psi_l, 0\right)$ and $\boldsymbol{\psi}_l^{(1)} = \left(0, \psi_l\right)$, as well as our choice that along these circles, the basis functions $\{\psi_l\}_{-M_r}^{M_r}$ are Fourier basis functions. Recall the interface conditions (3.3c) between two $\Omega_{j,1}$ and $\Omega_{j,2}$ and substitute the series representations of the Dirichlet and Neumann data from (3.30) into the interface conditions (3.3c):

$$\begin{aligned} \sum_{l=-M_r}^{M_r} \mathbf{c}_{5,l}^{(0)} \psi_l &= \sum_{l=-M_r}^{M_r} \mathbf{c}_{r,l}^{(0)} \psi_l, \\ \sum_{l=-M_r}^{M_r} \mathbf{c}_{5,l}^{(1)} \psi_l &= - \sum_{l=-M_r}^{M_r} \mathbf{c}_{r,l}^{(1)} \psi_l. \end{aligned}$$

By moving terms to the same side, factoring out basis functions, and exploiting the orthogonality of the Fourier basis functions, we obtain conditions for each $l = -M_r, \ldots, M_r$ that couple together systems (3.22) and (3.24):

$$\begin{aligned} \mathbf{c}_{5,l}^{(0)} - \mathbf{c}_{r,l}^{(0)} &= 0, \\ \mathbf{c}_{5,l}^{(1)} + \mathbf{c}_{r,l}^{(1)} &= 0. \end{aligned}$$

3.3 Numerical Results

In this section, numerical results are presented that demonstrate the performance of our method. The first part, Section 3.3.1, focuses on smaller, simpler examples that explore the discretization error of the method as well as the reflection error introduced by the absorbing boundary condition. Section 3.3.2 demonstrates more complex examples that are designed to more accurately represent a real PCRR. Demonstrative simulations are shown for linear waveguides (Section 3.3.2.1) as well as PCRR add/drop filters (Section 3.3.2.2).

For the results of this section, domains are constructed from collections of empty and rod subdomains (as discussed in previous sections), with empty subdomains handled as discussed in Chapter 2. The sides of each subdomain have a length of 2 and the rods have a radius of 0.37 (this selection is explained in Section 3.3.2). Every subdomain is discretized independently by grids consisting of $(n+1)\times(n+1)$ nodes, with $n = 2^m$, in order to take full advantage of the efficiency of the Fourier transform used in the AP solver.

Throughout these results, two kinds of errors are used to discuss convergence. First, we consider errors in the traditional sense:

$$||u^{(n)} - u||_\infty \to 0 \quad \text{as } n \to \infty. \tag{3.31}$$

In this case, $u^{(n)}$ refers to the solution computed using $n \times n$ grid nodes per subdomain and u is the true solution (when such a function can be derived). When a true solution cannot be derived, we instead assess the grid convergence by evaluating the norm of the difference between two succeeding approximate solutions obtained on a sequence of refined grids:

$$||u^{(n)} - u^{(2n)}||_\infty \to 0 \quad \text{as } n \to \infty. \tag{3.32}$$

Proper grid convergence in the sense of (3.31) clearly implies (3.32). Additionally, if (3.31) is characterized by a certain convergence rate, then the rate characterizing (3.32) will be at least as fast.

3.3.1 Preliminary Simulations

In order to evaluate the discretization error of the method and the reflection error introduced by the local ABC, an exact solution needs to be constructed for reference. For simplicity, the discretization and reflection errors are only computed directly in the rod subdomain for cases where the wavenumber is constant between the rod and the rest of the subdomain (i.e. $k_1 = k_2 = k$).

Consider the Green's function:

$$G(x,y) = \frac{i}{4} H_0^{(1)}(k\sqrt{x^2+y^2}) \tag{3.33}$$

where k is the wavenumber from (3.1a) and $H_0^{(1)}$ is the Hankel function. It is well-known that (3.33) satisfies the 2D Helmholtz equation with the point-source $f(x,y) = \delta(x,y)$ and the Sommerfeld radiation condition (1.7). However, the singularity at the origin of the Hankel function makes $G(x,y)$ ill-suited for numerical evaluation. Therefore, we instead consider a smoothed variation obtained by the following process:

1. Take the $x-$derivative of the fundamental solution ($G_x = \frac{\partial G}{\partial x}$). This will make the resulting

test solution anisotropic, i.e., more generic than a centrally symmetric solution.

2. Construct a function, $\eta(r)$, which is equal to 0 at $r=0$, equal to 1 when $r \geq R$, and is sufficiently smooth in $0 < r < R$. We use $R = \frac{1}{2}$ and require 5 vanishing derivatives at each endpoint, i.e. $\eta^{(d)}(0) = \eta^{(d)}(R) = 0$ for $d = 1, \dots, 5$.

3. Multiply the smoothing function by the differentiated fundamental solution to obtain a new solution $u = G_x \eta$.

This function $u_{rad} = G_x \eta$ is a solution to the homogeneous Helmholtz equation (i.e. $f \equiv 0$) outside the radius of $R = \frac{1}{2}$. Inside that radius, the source function f can be derived by plugging u_{rad} into the Helmholtz equation and taking its derivatives analytically:

$$f = \frac{\partial^2}{\partial x^2}(G_x \eta) + \frac{\partial^2}{\partial y^2}(G_x \eta) + k^2 G_x \eta. \tag{3.34}$$

For this section, the derivatives in (3.34) were computed with the (diff) command from the symbolic computation toolbox in MatLab. The function u_{rad} and its corresponding source function f can be seen in Figure 3.7, and u_{rad} is used throughout Sections 3.3.1.1 and 3.3.1.2 as a reference solution. The simulations of Sections 3.3.1.3 and 3.3.2 are driven by the same source term f from (3.34).

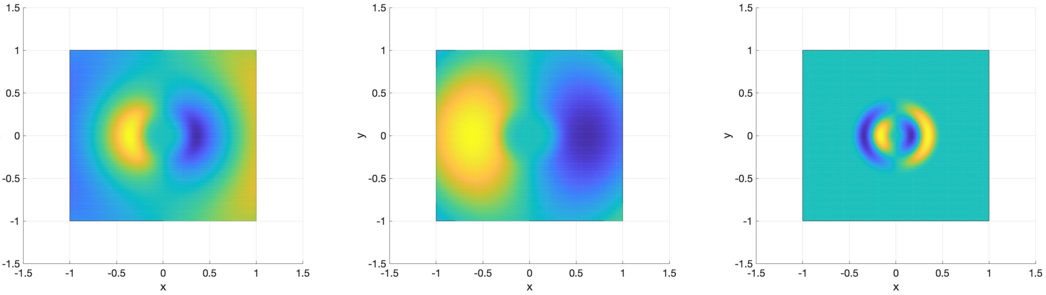


Figure 3.7 Real part (left) and imaginary part (center) of the function u_{rad}, and the real part of the corresponding derived source function f (right), all plotted for the wavenumber $k = 3$.

3.3.1.1 Discretization Error

To evaluate the discretization error introduced by the method, we compare the computed solution to the (exact) derived solution, $u_{rad} = G_x \eta$. This is done on a domain composed of an empty subdomain and a rod subdomain (see Figure 3.8), where the empty subdomain is centered about the origin. The Dirichlet boundary conditions and source function (3.34) are computed directly from the exact solution u_{rad}. Table 3.1 demonstrates that the method achieves the design fourth-order convergence rate of the underlying finite-difference method for a variety of wavenumbers.

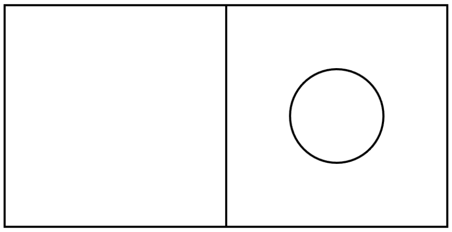


Figure 3.8 Depiction of the domain used in Table 3.1

Table 3.1 Discretization error $||u^{(n)} - u||_\infty$ and rate of convergence to the true solution $u = G_x\eta$ with $k = 5, 10$, and 15. Dirichlet boundary conditions and source function are derived from the solution.

	$k=5$		$k=10$		$k=15$	
n	Error	Rate	Error	Rate	Error	Rate
64	1.22e-04	-	9.14e-04	-	4.86e-01	-
128	7.53e-06	4.02	5.69e-05	4.01	2.11e-03	7.84
256	4.25e-07	4.15	3.65e-06	3.96	1.28e-04	4.04
512	2.83e-08	3.91	2.27e-07	4.00	8.01e-06	4.00
1024	1.64e-09	4.11	1.43e-08	3.99	5.01e-07	4.00

3.3.1.2 Reflection Error

Similar to the discretization error, evaluation of the reflection error requires an exact solution for comparison. The function u_{rad} serves as the exact solution here as it did in Section 3.3.1.1, and the source term is provided by (3.34). At the outer boundary, the second-order Engquist-Majda absorbing boundary conditions (1.8) are applied to approximate the outgoing waves.

The reflection error is a property of the chosen boundary conditions and therefore independent of the grid discretization. An example of this can be seen in the grid convergence analysis of Table 3.2, where the method appears to be converging as expected (fourth-order) on coarser grids before the convergence levels off. This is the point at which the reflection error dominates the discretization error. Due to the choice of a high-order finite-difference method, we generally expect this trade-off from discretization error to reflection error to happen at relatively coarse grids.

Table 3.2 Absolute reflection error and convergence rate for a 3 × 3 arrangement of empty subdomains with $k = 40$ (see Figure 3.9) and second-order Engquist-Majda absorbing boundary conditions along the exterior edges.

n	$\|\|u^{(n)} - u\|\|_\infty$	Rate
64	1.20e+02	-
128	4.38e+00	4.78
256	2.83e-01	3.95
512	1.76e-02	4.01
1024	1.76e-02	0.00
2048	1.76e-02	0.00

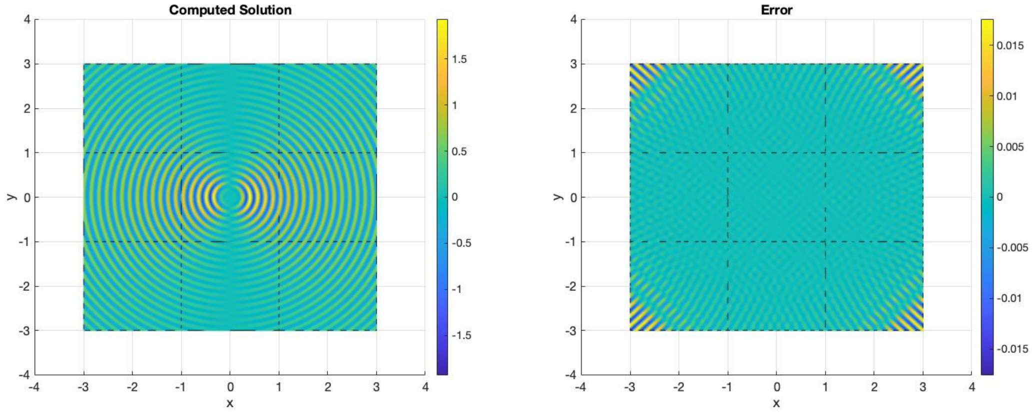


Figure 3.9 Real parts of the computed solution approximating u_{rad} with $k = 40$ (left) and the error between the computed solution and u_{rad} (right), corresponding to Table 3.2. The domain is a 3×3 arrangement of empty subdomains, and each subdomain is discretized by a grid of 512×512 nodes.

It is worth noting that the wavenumber chosen for Table 3.2 (and Figure 3.9) is higher than the other choices throughout Section 3.3. For smaller wavenumbers, the reflection error dominates the discretization error even at the coarser grid levels (see Table 3.3). In fact, [Engquist et al., 1977] provides the following approximation for the amplitude of reflected waves (denoted b) when using second-order Engquist-Majda absorbing boundary conditions:

$$b \approx a \left| \frac{\cos(\theta) - 1}{\cos(\theta) + 1} \right|^2 \tag{3.35}$$

where a is the amplitude of the incident wave and θ is the angle of incidence. The expression (3.35) is maximized when $\theta = \frac{\pi}{4}$, which occurs at the corners of the domain (visualized in Figure 3.9. With this choice of θ, (3.35) reduces to

$$b \approx a \left| \frac{\cos(\frac{\pi}{4}) - 1}{\cos(\frac{\pi}{4}) + 1} \right|^2 \implies b \approx (0.0294)a. \tag{3.36}$$

Roughly speaking, (3.36) tells us that the amplitude of the reflection error will be around 3% of the amplitude of the incident wave. To confirm this, the reflection error is evaluated across several domains for $k = 10$ and presented in Table 3.3. Note that the lack of convergence in Table 3.3 confirms that the reflection error is not dependent on the grid discretization. By comparing Table 3.3 with Table 3.1, it is clear that the discretization error for this solution is notably smaller than the reflection error. Further, note the improved performance as the size of the domain is increased. The reflection error is expected to decrease as the domain grows larger (and as the amplitude of the solution decreases), and this is confirmed in Table 3.3.

Tables 3.2 and 3.3 indicate that the reflection error prevents the computed solution from converging to the true solution, u_{rad}. The grid convergence, however, still takes place, albeit to a different solution that is subject to reflections from the artificial outer boundary due to boundary condition

Table 3.3 Maximum absolute and relative reflection errors along the exterior edges of Ω. Computed for $k = 10$ for domains constructed from several arrangements of empty subdomains (1×1, 3×3, and 5×5).

	1×1		3×3		5×5	
	Absolute	Relative	Absolute	Relative	Absolute	Relative
n	Error	Error	Error	Error	Error	Error
64	2.37e-02	3.80%	1.08e-02	3.30%	7.91e-03	2.89%
128	2.59e-02	4.16%	1.12e-02	3.42%	7.81e-03	2.85%
256	2.59e-02	4.16%	1.12e-02	3.43%	7.81e-03	2.86%
512	2.65e-02	4.25%	1.13e-02	3.45%	7.85e-03	2.87%

(3.25). This is corroborated by the data in Table 3.4 that shows convergence in the sense of (3.32).

Table 3.4 Convergence analysis with respect to the self-convergence metric from (3.32). Domains are composed of varying arrangements of empty subdomains (1×1, 3×3, and 5×5), and the wavenumber is $k = 10$.

	1×1		3×3		5×5	
n	$\|\|u^{(n)} - u^{(2n)}\|\|_\infty$	Rate	$\|\|u^{(n)} - u^{(2n)}\|\|_\infty$	Rate	$\|\|u^{(n)} - u^{(2n)}\|\|_\infty$	Rate
64	4.09e-04	-	4.45e-04	-	6.17e-04	-
128	2.65e-05	3.95	2.84e-05	3.97	3.69e-05	4.06
256	1.82e-06	3.86	1.70e-06	4.06	2.26e-06	4.03
512	1.04e-07	4.13	1.09e-07	3.97	1.43e-07	3.98

3.3.1.3 Inclusion of Scattering Rods

By allowing a change in the wavenumber between the rod and the rest of the subdomain (i.e. $k_1 \neq k_2$), the refraction of a signal between two mediums can be simulated. We generally only consider the case of $k_1 < k_2$ which represents the case where the rod has a higher refractive index than its surrounding material. This formulation makes it difficult to compose an exact solution for reference to demonstrate grid convergence, so convergence is considered as in (3.32). The domain is taken as in Figure 3.8 with the source function (3.34) centered about the empty subdomain and second-order Engquist-Majda absorbing boundary conditions along the exterior of the domain. The convergence rate in Table 3.5 is indicative of the design fourth-order convergence rate as discussed earlier and demonstrated in Tables 3.1 and 3.4.

3.3.2 PCRR Simulations

In this section, we present and discuss two forms of simulations associated with PCRRs. First is an isolated bus waveguide with a varying number of rows of scattering rods on either side of the channel. Second is the classic add-drop filter which can redirect a signal toward different output channels depending on the frequency.

Table 3.5 Convergence analysis with respect to the self-convergence metric from (3.32) for examples with scattering rods. The domain is depicted in Figure 3.8, with wavenumber jumps indicated from k_1 outside the rod to k_2 inside the rod.

	$k_1 = 1$		$k_1 = 3$	
	$k_2 = 3$		$k_2 = 10$	
n	$\|\|u^{(n)} - u^{(2n)}\|\|_\infty$	Rate	$\|\|u^{(n)} - u^{(2n)}\|\|_\infty$	Rate
64	2.82e-01	-	6.84e-01	-
128	3.33e-04	9.73	2.54e-03	8.07
256	1.44e-05	4.53	6.56e-06	8.60
512	1.29e-06	3.48	3.57e-07	4.20

As discussed in Section 3.3.1.2, the reflection error is expected to outweigh the discretization error, even at relatively coarse grids. Therefore, results are only reported on a single grid (indicated for each simulation). Additionally, we use a consistent setup scheme for determining the source function for each simulation (see Figure 3.10). The source function f (from (3.34)) is placed between rows of scattering rods, such that one empty subdomain separates the source's subdomain from the left edge of the domain. This way, the source can emulate an input signal that is primarily traveling from the left to the right.

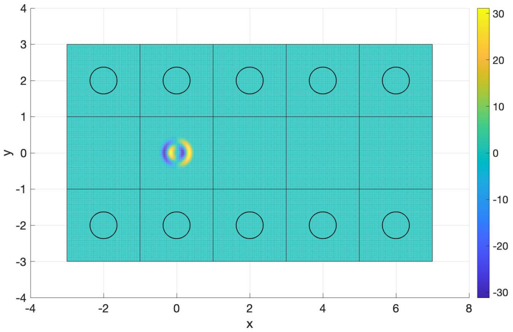


Figure 3.10 An example of how the source function (3.34) is placed within a PCRR domain. The source is centered over one of the empty subdomains such that one empty subdomain sits between the source's support and the exterior boundary of the domain.

The parameters for the simulations are chosen to reflect those used in [Chremmos et al., 2010]. The descriptions of the primary parameters are as follows:

- a = Lattice periodicity, or the shortest center-to-center distance between two rods. Consequently, a is also the length of each side of the subdomains.

- r = Radius of the scattering rods. The ratio of the radius of the rods to the lattice periodicity (i.e. r/a) is an important factor in determining valid frequencies for the PCRR.

- n_{air}, n_{rod} = Refractive indices of the background (air) and scattering rods, respectively.

In [Chremmos et al., 2010], the lattice periodicity is chosen as $a = 540$nm and the radius of the rods is $r = 100$nm, resulting in a ratio of $r/a \approx 0.185$. The refractive indices are selected to represent air ($n_{air} = 1.00$) and silicon ($n_{rod} = 3.48$). These parameters correspond to bus waveguides that support wavelengths between approximately $\lambda = 1270$nm and $\lambda = 1780$nm.

For the simulations in this chapter, the parameters have been nondimensionalized. The lattice periodicity (and side length of subdomains) is 2. In order to maintain the ratio of the rod radius to lattice periodicity, we take $r = 0.37 = 2 \times 0.185$ (hence the choice for the rod radius in the results of Section 3.3.1). The refractive indices provide a ratio for the wavenumbers we select, namely $k_2 = 3.48 \times k_1$. The background wavenumber k_1 can then be derived by equating the ratio of the lattice periodicity to the desired wavelength from each example, i.e.

$$\begin{aligned} \text{(Physical)} \quad \frac{a}{\lambda} &= \frac{k_1 a}{2\pi} \quad \text{(Nondimensionalized)} \\ \frac{540}{\lambda} &= \frac{2k_1}{2\pi} \\ \frac{540\pi}{\lambda} &= k_1. \end{aligned} \tag{3.37}$$

Therefore, a physical wavelength of 1550nm approximately corresponds to a wavenumber of $k_1 = 1.08$. As our discretization is fourth order accurate, for this low value of the wavenumber the error due to the second order Engquist-Majda boundary condition becomes dominant already on fairly coarse grids, see Section 3.3.1.2.

For most of the figures in this section, we have chosen to present the imaginary part of the solution. The real part of the source function has a high degree of variation at the origin, causing the real part of the resulting solution to have abnormally high variation near the source. However, the imaginary part changes much more gradually near the source, forming a much smoother and visually interpretable image while still demonstrating the same qualitative behaviors as its corresponding real part.

3.3.2.1 Bus Waveguides

Recall that a ring resonator is constructed from two bus waveguides and a ring-shaped waveguide (Figure 1.1). For the specific case of a PCRR (Figure 1.2) the bus waveguide can be isolated to make its own domain as in Figure 3.11. From [Chremmos et al., 2010], it is known that the number of rows of scattering rods on either side of the open channel largely affects how efficiently the channel propagates waves. Generally speaking, increasing the number of rows of rods will decrease the amount of leakage from the waveguide at the expense of more computation time to account for more rods. By computing the solution in domains with varying numbers of buffer rows of rods, we can observe how the solution changes to determine how many rows are needed to capture the qualitative behavior.

In order to analyze the impact of buffer rows of rods, we assemble a sequence of simulations that represent bus waveguides with varying numbers of buffer rows. Each problem uses the same source

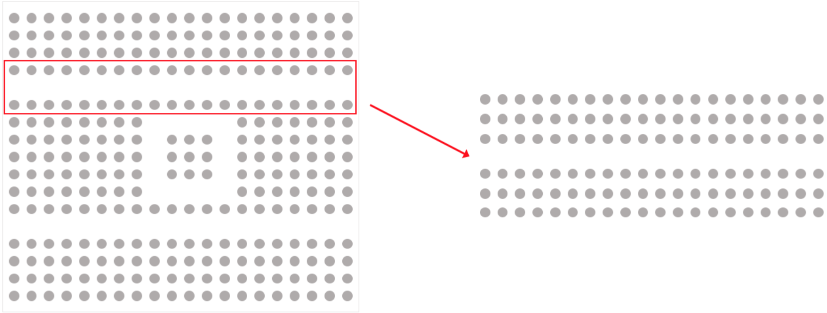


Figure 3.11 One of the bus waveguides can be isolated and considered as its own domain.

function from (3.34) placed in the waveguide as described in Figure 3.10, along with second-order Engquist-Majda boundary conditions along the exterior edges of the relevant subdomains. The domain of interest is a 1×15 assembly of empty subdomains with the corresponding number of buffer rows of rod subdomains attached to both the top and bottom. The resulting computed solutions are collectively presented on the left side of Figure 3.12. Note that the plots in Figure 3.12 only represent the empty subdomains down the center of the domain.

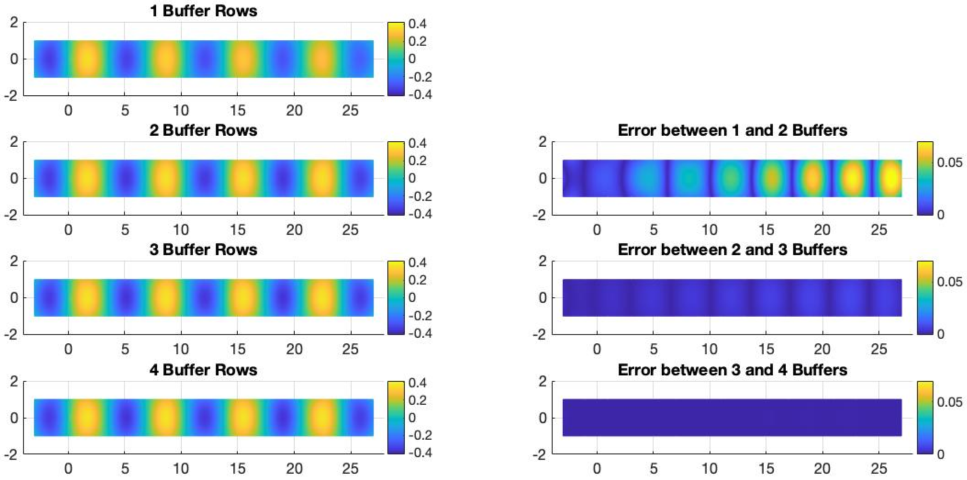


Figure 3.12 (Left) Imaginary part of the solution for varying numbers of buffer rows of rods and (right) the absolute difference between a given solution and the preceding case. Plotted for $k = 1.229$ on a 129×129 grid for each subdomain.

The right side of Figure 3.12 shows how the solution changes after each addition of a new buffer row of rod subdomains. The error is calculated as the absolute difference between two solutions within the empty subdomains, where the solution most clearly depicts a traveling wave. Notice that the signal is noticeably fuzzy in the first solution (top-left of Figure 3.12), but becomes sharper when a second row of buffer rods is included. This difference is visualized in the first image of the right column, where it can clearly be seen that the difference between the two solutions grows toward the end of the domain. Although there is little visual difference between the cases of 2 and 3 buffers in the left column, the second error plot still shows a discernible change at the right end of the channel. Finally, the change when moving from 3 to 4 buffer rows is even less than the previous case, and visually negligible when compared to the previous cases.

Ideally, the number of buffer rows would be taken to infinity (or at least arbitrarily high) to provide the best approximation, but this is obviously impractical. Referring to the case of 2 buffers, $||u||_\infty \approx 0.3256$ and the maximum error between the cases of 2 and 3 buffers is approximately 0.0067. Adding an extra row of buffers only improves the accuracy by 2%. Given that the reflection error from the ABCs is around 3%, 2 buffer layers are chosen for the larger simulations of the next section.

3.3.2.2 PCRR Add-Drop Filter

The final set of simulations concern the functionality of a 2D PCRR add-drop filter. The only new feature introduced for this structure is the ring resonator itself. The resonator is formed in a similar fashion as the bus waveguides from Section 3.3.2.1, with the difference that the channel formed by removing rods is now a square (note Figures 1.2, 1.3, and 3.11). Note that the shape of the ring will depend on the symmetry of the lattice of rods; alternative arrangements (such as a hexagonal resonator) can be found in [Chremmos et al., 2010].

Following the square lattice example from [Chremmos et al., 2010], a PCRR is constructed around a resonator with a 3×3 set of rods at its center (as in Figure 3.11). Bus waveguides are appended to the top and bottom of the resonator with one set of rods separating the bus waveguides from the resonator. One extra row of rods is included on the top and bottom of the domain to prevent leakage (following the example set by Section 3.3.2.1) forming a 13×15 lattice of subdomains as in Figure 3.13. The system is driven by the source function f from (3.34), placed as in Figure 3.10, which simulates an input signal from the top-left port with a given frequency. The frequency is determined by the wavenumber k in the corresponding Helmholtz equation, and the relationship of this k to a physical wavelength is given in (3.37).

The main simulations of focus are presented in Figures 3.14 and 3.15. They represent two wavenumbers with qualitatively different behaviors in the PCRR. The first simulation, Figure 3.14, represents a signal that is designed to pass through the device without being filtered. Some energy does spread throughout the domain, particularly through the channels, but this kind of leakage is to be expected given the use of discrete rods in place of continuous walls. The important piece, qualitatively speaking, is that the strength of the signal in the bottom bus waveguide is distinctly lower than in the top bus waveguide. For the second simulation, Figure 3.15 represents a signal that the PCRR is designed to filter through the ring resonator. It is demonstrating the expected behavior, which is for the strength of the signal to be greater at the bottom-left port than either port on the right side of the domain.

Figures 3.16 and 3.17 provide examples of some of the alternative behaviors that the PCRR can

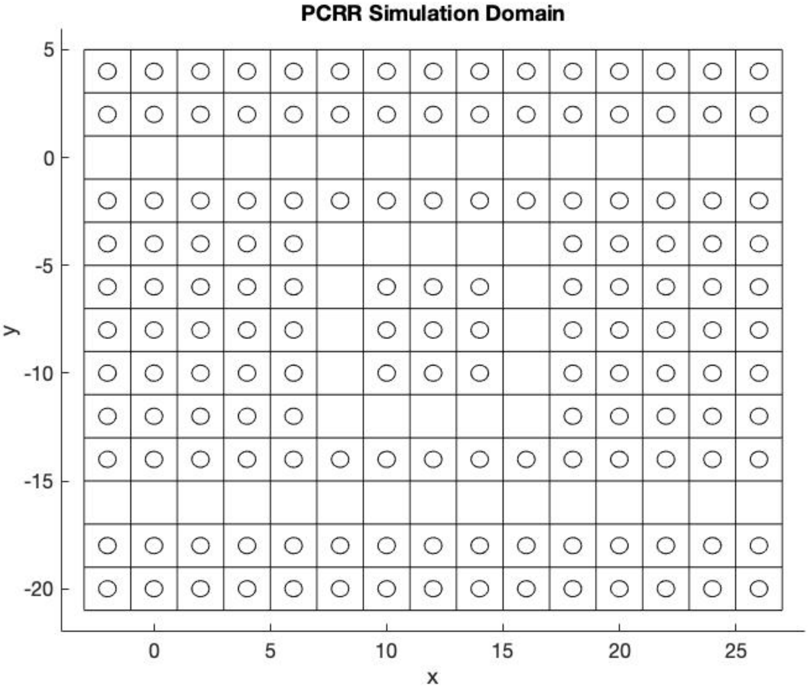


Figure 3.13 The domain used for the PCRR simulations of Section 3.3.2.2. Solid lines denote interfaces between subdomains, and the source function is centered about the origin.

express as the wavenumber/wavelength go further outside of the intended window of use for the parameters. In Figure 3.16, the signal drops from the top bus into the resonator, but not into the bottom bus. In Figure 3.17, the signal is approximately as strong at the right end of the top bus as it is at the left end, but it enters the resonator just as in Figure 3.16. These two examples demonstrate apparent deficiencies, which we believe to be attributed to the model as opposed to the method. A known problem for such PCRRs stems from the the fact that a square duct is used in place of a circle in the true ring resonator case, causing certain degenerative behaviors [Chremmos et al., 2010]. The known way to combat this is by introducing four additional scattering rods in the corners of the resonator such that they are equidistant from the three rods that currently form the corner. Unfortunately, these points are at the intersection of the building blocks, making this supplement incompatible with the framework in its current form. One potential solution is discussed in Chapter 4.

3.4 Conclusions

The goal of this chapter is to expand the framework introduced in Chapter 2 and [North et al., 2022] for solving domain decomposition problems with the MDP, and to demonstrate the method's ability to simulate 2D photonic crystal ring resonators. To this end, the "rod subdomain" was introduced and the implementation of local absorbing boundary conditions was described using the second-order

Engquist-Majda conditions. Analysis of the discretization and reflection errors corroborated the design properties of the underlying finite-difference method and ABCs, and numerical simulations demonstrated that the method is able to properly capture the intended behavior of the PCRR.

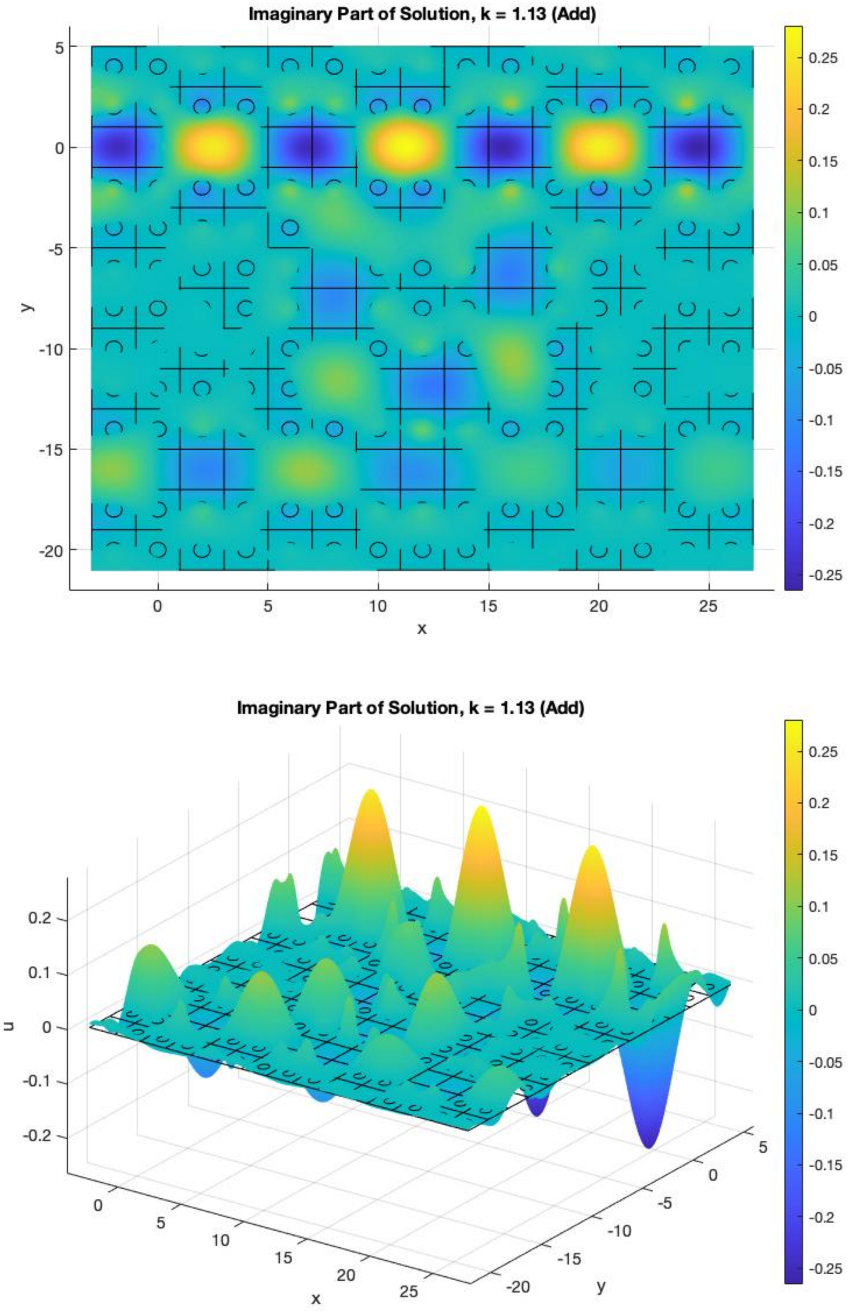


Figure 3.14 Example of a PCRR operating at an add-frequency corresponding to the wavenumber $k = 1.13$. Each subdomain is discretized by a grid of 129×129 nodes.

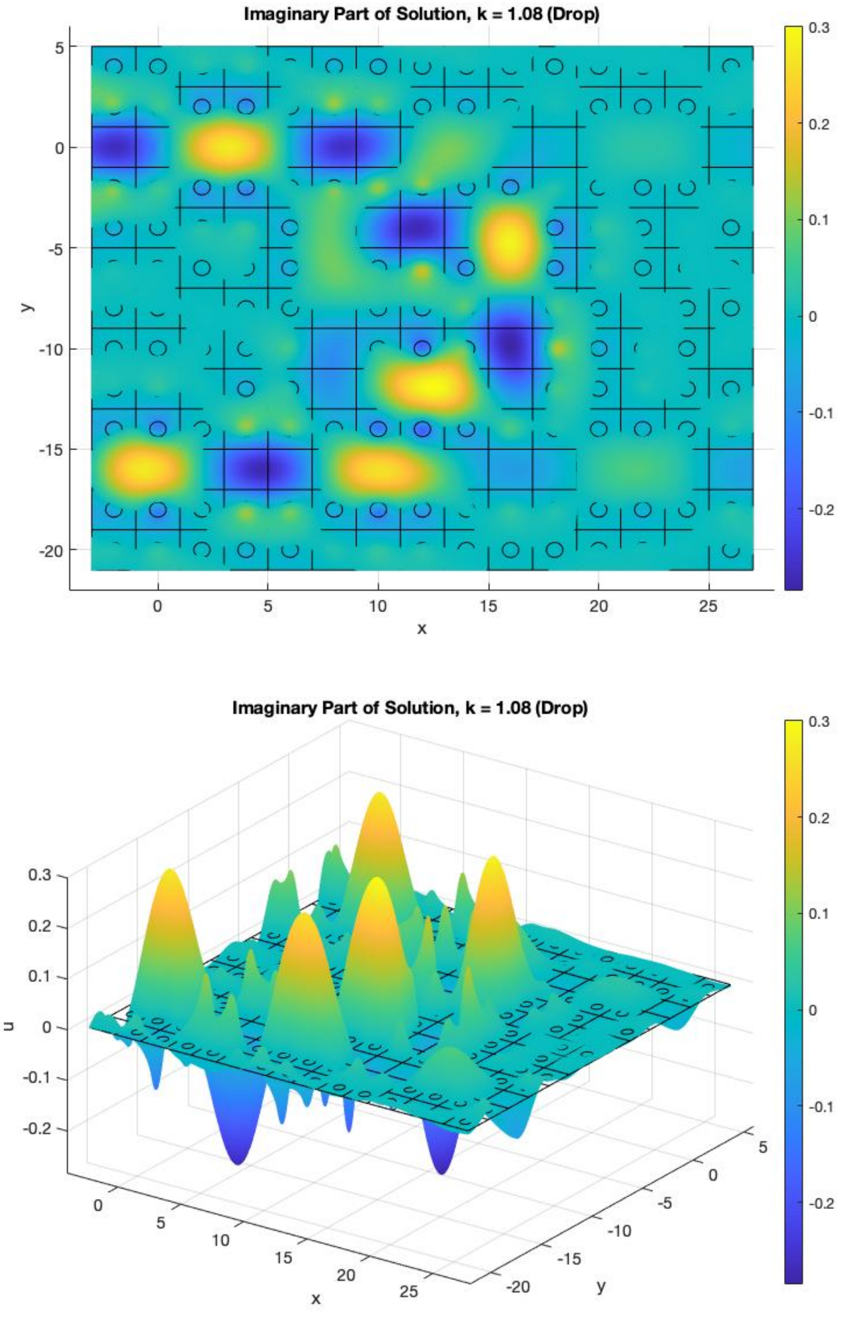


Figure 3.15 Example of a PCRR operating at a drop-frequency corresponding to the wavenumber $k = 1.08$. Each subdomain is discretized by a grid of 129×129 nodes.

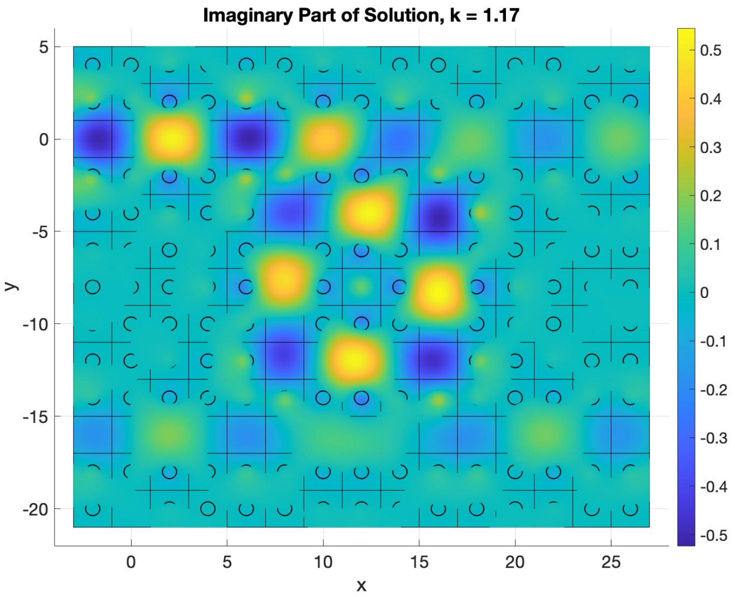


Figure 3.16 Example of a PCRR operating at a resonant frequency corresponding to the wavenumber $k = 1.17$.

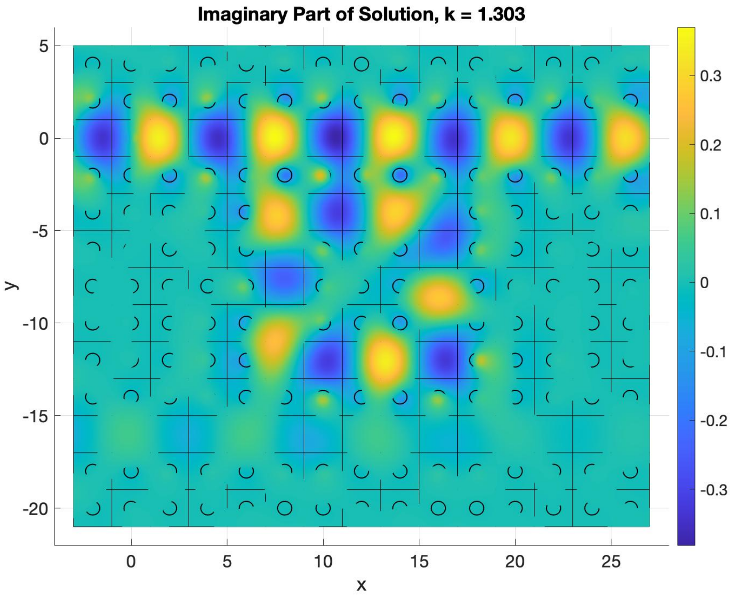


Figure 3.17 Example of a PCRR operating at a resonant frequency corresponding to the wavenumber $k = 1.303$.

CHAPTER

4 FUTURE DIRECTIONS

The ideas described in this dissertation are only a framework from which one can build in a number of directions. Due to this, we suggest a variety of different directions for potential future research.

As the method stands, the rod building block is a composition of two subdomains: the interior of the circle and the area between the square and the circle. Because of this, including one rod building block increases the computational cost of the method more than adding an empty building block. Instead, we believe it could be beneficial to incorporate the scattering rod directly into the AP of the building block. The interface conditions along the rod would no longer be handled at run-time, reducing the cost of the overall QR-factorization. Intuitively speaking, our method currently uses the same AP for all of the subdomains, but this change would use a customized AP for each unique building block.

Given the “building block” nature of the framework, a natural question is whether the building blocks can be combined into larger macro-blocks. The main difficulty in using blocks of different sizes is getting the edges to align so that the orthogonality of the basis functions can be exploited in the interface conditions as in (2.30). By combining the existing blocks into larger macro-blocks, arbitrarily large sections of a simulation’s domain can be stored in pre-computed operators so that the interface conditions are resolved before run-time. Of course, this would lead to a higher storage cost than just having two fundamental blocks, so the benefit of this approach would have to be considered based on the application.

If stronger performance in PCRR simulations is desired, then a valuable change to consider is finding a way to implement supplemental rods that do not fall in the center of building blocks (see Figure 4.1). In [Chremmos et al., 2010] these are shown to substantially improve the spectral selectivity of the PCRR. The framework as presented in this dissertation does not support — at least not in an obvious way — adding such supplemental scattering rods into the domain. We believe that one of the ways discussed above regarding macro-blocks could hold promise, allowing a larger block to be defined that simply contains multiple scattering rods in the appropriate locations. In Figure 4.1, this is demonstrated by the example block composed of four existing blocks. The dotted lines imply the interfaces that exist in our current decomposition, but this will be incompatible with a scattering rod at the center.

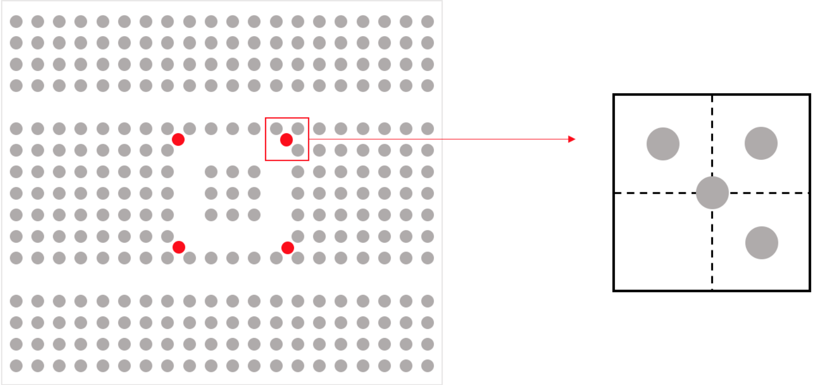


Figure 4.1 Example of how extra rods can be added to a PCRR setup to improve performance. On the left, the extra rods are in red and are placed equidistant from the three closes existing rods. On the right, the dotted lines represent what would be interfaces in the existing framework.

Another direction for the building blocks themselves is to expand into different shapes. For example, PCRRs can be constructed from various symmetries [Chremmos et al., 2010], so another viable building block shape could be hexagons. Moving beyond the cases of PCRRs, having a small collection of unique shapes could provide a great deal of flexibility in the kinds of domains one could create decompositions for. However, it is worth noting that, like the examples chosen for numerical demonstrations in Sections 2.2 and 3.3, greater benefit is expected from domains that are more easily tiled with as few unique building blocks as possible. Regarding the choice of APs and other details for implementing the MDP on domains of general shapes, [Medvinsky et al., 2016] and [D. S. Britt et al., 2013] are useful resources with which to start.

Many choices in the implementation of the method were made for convenience and the method's computational cost could be improved by giving careful consideration to several of these choices. Most of these choices are specific to the construction and factorization of the overall linear system (2.32). For example, by allowing the number of basis functions along the edges of subdomains to vary, one can adaptively select them according to the expected complexity of the solution in that part of the domain. By only using the necessary number of basis functions, the width of the final linear system that gets factorized can be reduced, potentially by a significant amount. In a similar sense, discretization grids can be chosen at different degrees of refinement depending on the nature of the solution in given parts of the domain.

BIBLIOGRAPHY

Albright, J., Epshteyn, Y., Medvinsky, M. & Xia, Q. (2017). "High-order numerical schemes based on difference potentials for 2D elliptic problems with material interfaces". *Appl. Numer. Math.* **111**, pp. 64–91.

Babuška, I. M. & Sauter, S. A. (2000). "Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?" *SIAM Rev.* **42**.3. Reprint of SIAM J. Numer. Anal. **3**4 (1997), no. 6, 2392–2423 [MR1480387 (99b:65135)], pp. 451–484.

Bayliss, A., Goldstein, C. I. & Turkel, E. (1985). "On accuracy conditions for the numerical computation of waves". *J. Comput. Phys.* **59**.3, pp. 396–404.

Bayliss, A., Goldstein, C. I. & Turkel, E. (1983). "An iterative method for the Helmholtz equation". *J. Comput. Phys.* **49**.3, pp. 443–457.

Bayliss, A., Gunzburger, M. & Turkel, E. (1982). "Boundary conditions for the numerical solution of elliptic equations in exterior regions". *SIAM J. Appl. Math.* **42**.2, pp. 430–451.

Bayliss, A. & Turkel, E. (1980). "Radiation boundary conditions for wave-like equations". *Comm. Pure Appl. Math.* **33**.6, pp. 707–725.

Boubendir, Y., Antoine, X. & Geuzaine, C. (2012). "A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation". *J. Comput. Phys.* **231**.2, pp. 262–280.

Britt, D. S., Tsynkov, S. V. & Turkel, E. (2013). "A high-order numerical method for the Helmholtz equation with nonstandard boundary conditions". *SIAM J. Sci. Comput.* **35**.5, A2255–A2292.

Britt, S., Tsynkov, S. & Turkel, E. (2011). "Numerical simulation of time-harmonic waves in inhomogeneous media using compact high order schemes". *Commun. Comput. Phys.* **9**.3, pp. 520–541.

Calderón, A.-P. (1963). "Boundary value problems for elliptic equations". *Outlines Joint Sympos. Partial Differential Equations (Novosibirsk, 1963)*. Acad. Sci. USSR Siberian Branch, Moscow, pp. 303–304.

Chremmos, I., Uzunoglu, N. K. & Schwelb, O. (2010). *Photonic microresonator research and applications*. Springer.

Deraemaeker, A., Babuška, I. & Bouillard, P. (1999). "Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions". *International Journal for Numerical Methods in Engineering* **46**.4, pp. 471–499.

Després, B. (1993). "Domain decomposition method and the Helmholtz problem. II". *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993)*. SIAM, Philadelphia, PA, pp. 197–206.

Dolean, V., Jolivet, P. & Nataf, F. (2015). *An introduction to domain decomposition methods*. Algorithms, theory, and parallel implementation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, pp. x+238.

Driscoll, T., Hale, N. & Trefethen, L. (2014). *Chebfun Guide*. Pafnuty Publications, Oxford.

Engquist, B. & Majda, A. (1977). "Absorbing boundary conditions for the numerical simulation of waves". *Math. Comp.* **31**.139, pp. 629–651.

Fard, S. T., Grist, S. M., Donzella, V., Schmidt, S. A., Flueckiger, J., Wang, X., Shi, W., Millspaugh, A., Webb, M., Ratner, D. M., Cheung, K. C. & Chrostowski, L. (2013). "Label-free silicon photonic biosensors for use in clinical diagnostics". *Silicon Photonics VIII*. Ed. by Kubby, J. & Reed, G. T. Vol. 8629. International Society for Optics and Photonics. SPIE, pp. 49–62.

Gander, M. J. (2008). "Schwarz methods over the course of time". *Electron. Trans. Numer. Anal.* **31**, pp. 228–255.

Gander, M. J. & Santugini, K. (2016). "Cross-points in domain decomposition methods with a finite element discretization". *Electron. Trans. Numer. Anal.* **45**, pp. 219–240.

Gordon, D. & Gordon, R. (2020). "CADD: a seamless solution to the Domain Decomposition problem of subdomain boundaries and cross-points". *Wave Motion* **98**, pp. 102649, 11.

Hagstrom, T. & Hariharan, S. I. (1988). "Accurate boundary conditions for exterior problems in gas dynamics". *Math. Comp.* **51**.184, pp. 581–597.

Harari, I. & Turkel, E. (1995). "Accurate finite difference methods for time-harmonic wave propagation". *J. Comput. Phys.* **119**.2, pp. 252–270.

Hesthaven, J. S., Gottlieb, S. & Gottlieb, D. (2007). *Spectral Methods for Time-Dependent Problems*. Vol. 21. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, pp. x+273.

Karageorghis, A. (1988). "A note on the Chebyshev coefficients of the general order derivative of an infinitely differentiable function". *J. Comput. Appl. Math.* **21**.1, pp. 129–132.

Lions, P.-L. (1990). "On the Schwarz alternating method. III. A variant for nonoverlapping subdomains". *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*. SIAM, Philadelphia, PA, pp. 202–223.

Magura, S., Petropavlovsky, S., Tsynkov, S. & Turkel, E. (2017). "High-order numerical solution of the Helmholtz equation for domains with reentrant corners". *Appl. Numer. Math.* **118**, pp. 87–116.

Medvinsky, M., Tsynkov, S. & Turkel, E. (2012). "The method of difference potentials for the Helmholtz equation using compact high order schemes". *J. Sci. Comput.* **53**.1, pp. 150–193.

— (2013). "High order numerical simulation of the transmission and scattering of waves using the method of difference potentials". *J. Comput. Phys.* **243**, pp. 305–322.

Medvinsky, M., Tsynkov, S. & Turkel, E. (2016). “Solving the Helmholtz equation for general smooth geometry using simple grids”. *Wave Motion* **62**, pp. 75–97.

Modave, A., Royer, A., Antoine, X. & Geuzaine, C. (2020). “An optimized Schwarz domain decomposition method with cross-point treatment for time-harmonic acoustic scattering”. *HAL- 02432422*.

North, E., Tsynkov, S. & Turkel, E. (2022). “Non-iterative domain decomposition for the Helmholtz equation with strong material discontinuities”. *Appl. Numer. Math.* **173**, pp. 51–78.

— (n.d.). “Simulation of photonic crystal ring resonators using domain decomposition and difference potentials”. Under review.

Reznik, A. A. (1982). “Approximation of surface potentials of elliptic operators by difference potentials”. *Soviet Math. Dokl.* **25**.2, pp. 543–545.

— (1983). “Approximation of the Surface Potentials of Elliptic Operators by Difference Potentials and Solution of Boundary-Value Problems (in Russian)”. PhD thesis. Moscow: Moscow Institute of Physics and Technology.

Rodriguez-Esquerre, V., Koshiba, M. & Hernandez-Figueroa, H. (2005). “Finite-element analysis of photonic Crystal cavities: time and frequency domains”. *Journal of Lightwave Technology* **23**.3, pp. 1514–1521.

Romero-Vivas, J., Chigrin, D. N., Lavrinenko, A. V. & Torres, C. M. S. (2005). “Resonant add-drop filter based on a photonic quasicrystal”. *Opt. Express* **13**.3, pp. 826–835.

Rubio-Merccdes, C., Hernandez-Figueroa, H., Rodriguez-Esquerre, V. & Davanco, M. (2004). “Finite-element frequency-domain analysis of 2D photonic crystal resonant cavities”. *Proceedings of the 4th International Conference on Numerical Simulation of Optoelectronic Devices, 2004. NUSOD ’04.* Pp. 111–112.

Ryaben’ kii, V. S. (2002). *Method of difference potentials and its applications*. Vol. 30. Springer Series in Computational Mathematics. Translated from the 2001 Russian original by Nikolai K. Kulman. Springer-Verlag, Berlin, pp. xviii+538.

Ryaben’kii, V. S. (1985). “Boundary equations with projections”. *Russian Mathematical Surveys* **40**.2, pp. 147–183.

Schwarz, H. A. (1870). “Über einen Grenzübergang durch alternierendes Verfahren”. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich* **15**, pp. 272–286.

Seeley, R. T. (1966). “Singular integrals and boundary value problems”. *Amer. J. Math.* **88**, pp. 781–809.

Singer, I. & Turkel, E. (1998). “High-order finite difference methods for the Helmholtz equation”. *Comput. Methods Appl. Mech. Engrg.* **163**.1-4, pp. 343–358.

— (2006). “Sixth-order accurate finite difference schemes for the Helmholtz equation”. *J. Comput. Acoust.* **14**.3, pp. 339–351.

Sommerfeld, A. (1949). *Partial Differential Equations in Physics*. Translated by Ernst G. Straus. Academic Press, Inc., New York, N.Y., pp. xi+335.

Tsynkov, S. V. (1998). “Numerical solution of problems on unbounded domains. A review”. *Appl. Numer. Math.* **27**.4. Absorbing boundary conditions, pp. 465–532.

Turkel, E., Gordon, D., Gordon, R. & Tsynkov, S. (2013). “Compact 2D and 3D sixth order schemes for the Helmholtz equation with variable wave number”. *J. Comput. Phys.* **232**, pp. 272–287.