Applied Numerical Mathematics 173 (2022) 51–78

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Non-iterative domain decomposition for the Helmholtz equation with strong material discontinuities ☆

Evan North [a], Semyon Tsynkov [a,*], Eli Turkel [b]

[a] Department of Mathematics, North Carolina State University, Box 8205, Raleigh, NC 27695, USA
[b] School of Mathematical Sciences, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel

ARTICLE INFO

Article history:
Received 5 August 2021
Received in revised form 12 October 2021
Accepted 26 October 2021
Available online 18 November 2021

Keywords:
Time-harmonic waves
Non-overlapping domain decomposition
Calderon's operators
Difference potentials
Exact coupling between subdomains
High-order accuracy
Spectral representation at the boundary
Compact finite difference schemes
Direct solution
Complexity bounds
Discontinuous coefficients
Interior cross-points

ABSTRACT

Many wave propagation problems involve discontinuous material properties. We propose to solve such problems by non-overlapping domain decomposition combined with the method of difference potentials (MDP). The MDP reduces the Helmholtz equation on each subdomain to a Calderon's boundary equation with projection on the 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 direct solver. Our method proves to be insensitive to large jumps in the wavenumber for transmission problems, as well as interior cross-points and mixed boundary conditions, which may be a challenge to many other domain decomposition methods.

1. Introduction

The Helmholtz equation governs the propagation of time-harmonic waves. In many applications, the propagation medium involves material discontinuities, i.e., surfaces across which the wave speed or equivalently, the wavenumber, undergoes a jump. We propose to solve such problems by non-overlapping domain decomposition methods (DDMs), where material discontinuities coincide with interfaces between subdomains. The rationale is that the solution to the Helmholtz equation loses regularity at the surfaces with material discontinuities, and to maintain existence and uniqueness, the PDE itself must be accompanied by additional interface conditions. Most frequently, the latter require continuity of the solution and its flux across the interface (although many other choices are available). At the same time, the core idea of the DDMs is to partition the domain into smaller, simpler subdomains thus creating subproblems coupled to one another along their interfaces. The coupling is rendered via the additional conditions, called transmission conditions, that approximate the true interface

☆ Work supported by the US Army Research Office (ARO) under grant W911NF-16-1-0115 and the US–Israel Binational Science Foundation (BSF) under grants 2014048 and 2020128.

* Corresponding author.
E-mail addresses: einorth@ncsu.edu (E. North), tsynkov@math.ncsu.edu (S. Tsynkov), turkel@tauex.tau.ac.il (E. Turkel).
URLs: https://stsynkov.math.ncsu.edu (S. Tsynkov), http://www.math.tau.ac.il/~turkel/ (E. Turkel).

https://doi.org/10.1016/j.apnum.2021.10.024
0168-9274/

conditions required by the PDE. One may therefore expect that the DDMs will provide a framework for accommodating the material discontinuities along the interfaces.

The original development of DDMs for the numerical solution of PDEs did not focus on material discontinuities. It was rather motivated by the need for parallel implementation in order to alleviate the growth of the computational cost for large propagation domains. Coupling between subdomains is resolved in DDMs 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 [13,28,38]. The convergence rate of this iterative process is heavily dependent on the transmission conditions, the choice of which is still a highly active research area [4,22,37]. Typically, accounting for the global behavior in the boundary update step — where the transmission conditions are utilized — leads to more expensive updates but fewer iterations for convergence. On the other hand, localized approximations in the transmission conditions tend to result in faster updates but more iterations.

Material discontinuities may adversely impact the convergence of DDM iterations and thus present an additional challenge for the numerical solution of the Helmholtz equation. This is especially true when the jumps in the wavenumber are large (strong discontinuities) and/or when several interfaces meet at one point. The latter scenario is encountered frequently and referred to as interior cross-points in the DDM literature. Strong discontinuities and interior cross-points require a special case-by-case treatment in the traditional DDM framework, see, e.g., [9,10,16,18,25].

In this paper, we address the issue of coupling between subdomains by imposing the exact interface conditions and circumventing the iterative process altogether. In addition to letting the wavenumber be discontinuous across interfaces, we allow the boundary conditions on different segments of the boundary to have different type (mixed boundary conditions). The global behavior of the solution is accounted for by enforcing the continuity of the solution and its first normal derivative at the interfaces, while solutions on individual subdomains are approximated with high order accuracy and computed concurrently using a direct solver.

Our approach relies on the method of difference potentials (MDP). Originally proposed by Ryaben'kii [30,31], the MDP can be interpreted as a discrete version of the method of Calderon's operators [8,34] in the theory of partial differential equations. The MDP reduces a given PDE from its domain to the boundary. The resulting boundary formulation involves an operator equation (Calderon's boundary equation with projection (BEP)) with both Dirichlet and Neumann data in the capacity of unknowns. Once the BEP has been solved, the solution on the domain is reconstructed using Calderon's potential. Therefore, the MDP allows one to parameterize solutions on the domain using their boundary data. This proves very 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 [24]. This yields an overall linear system to be solved only at the combined boundary.

The computation of the discrete Calderon's boundary projections is performed ahead of time and in parallel. In the simplest case where all subdomains have the same shape and the wavenumber is uniform, the operators are computed for one subdomain and then reused. If the wavenumber jumps between subdomains and/or the subdomains differ in shape, then additional operators need to be computed. However, new boundary conditions, including mixed, do not require that Calderon's projections be recomputed [6]. As the method enforces the proper interface conditions exactly, its performance is not affected by the presence of even strong material discontinuities and/or interior cross-points, which is in sharp distinction with many traditional DDMs. The structure of the decomposition, e.g., strip-type vs. checkerboard-type, does not affect the performance either.

In [15], a domain decomposition approach based on difference potentials was applied to solving a chemotaxis model; it enabled adaptive grid refinement in one of the two subdomains. Other non-iterative domain decomposition methods include [26,27], where the decomposition is applied to a finite element discretization of the vector Helmholtz equation.

The outline of this paper is as follows: In Section 2, we introduce DDMs for the Helmholtz equation. Section 3 establishes the representative subdomain and covers the necessary information to implement the MDP, ending with the modifications necessary to combine the MDP with DDM. Details for a practical implementation are outlined in Section 3.5 and the complexity of the method is discussed in Section 3.6. In Section 4, numerical results are presented to validate the algorithm, corroborate the claims of complexity from Section 3.6, and explore the practical limits of the method. Finally, in Section 5 we provide a summary and propose directions for future research.

2. Domain decomposition

Domain decomposition methods were first introduced by Schwarz [33] to prove the existence and uniqueness of solutions to the Poisson equation over irregularly shaped domains. The original Schwarz algorithm used overlapping decompositions (Fig. 1), but was later extended to non-overlapping decompositions (Fig. 2) by Lions [20]. In this paper, we focus on non-overlapping subdomains. Accordingly, we begin with 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 [13,17,28, 38].

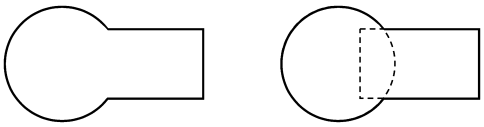
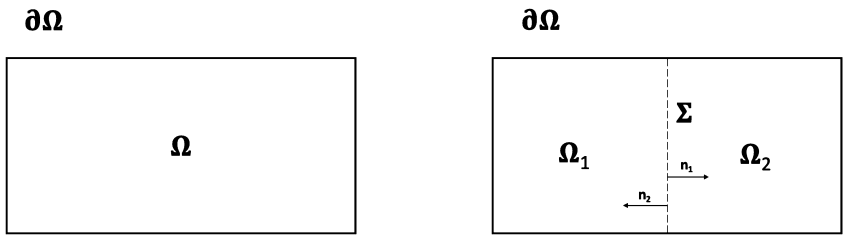


Fig. 1. The classical DDM example. An irregular domain composed of two shapes, decomposed into two overlapping subdomains.



(a) Rectangular domain. (b) Two-domain decomposition.

Fig. 2. 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.

2.1. Non-overlapping formulation

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

$$\begin{cases} \Delta u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{1}$$

Consider a partitioning of Ω that splits the domain into two subdomains, Ω_1 and Ω_2, by introducing an artificial interface $\Sigma = \overline{\Omega}_1 \cap \overline{\Omega}_2$ as in Fig. 2b. The BVP (1) can be reformulated over the new subdomains individually:

$$\begin{cases} \Delta u_1 = f & \text{in } \Omega_1 \\ u_1 = 0 & \text{on } \overline{\Omega}_1 \cap \partial\Omega \end{cases} \tag{2a}$$

$$\begin{cases} \Delta u_2 = f & \text{in } \Omega_2 \\ u_2 = 0 & \text{on } \overline{\Omega}_2 \cap \partial\Omega \end{cases} \tag{2b}$$

$$\begin{cases} u_1 = u_2, & \text{on } \Sigma \\ \frac{\partial u_1}{\partial \mathbf{n_1}} = -\frac{\partial u_2}{\partial \mathbf{n_2}} & \text{on } \Sigma \end{cases} \tag{2c}$$

where the interface conditions (2c) guarantee that the combined solution of (2) coincides with that of (1):

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

Conditions other than (2c) can be formulated on Σ so that the resulting combined problem is well-posed, but its solution will be different from the solution of (1).

There are two separate interface conditions in (2c). They apply to both subproblems (2a) and (2b) at the same time and couple them together. However, each subproblem (2a) or (2b) considered independently, i.e., with no connection to the other one, 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 (2c), one cannot specify more than one boundary condition on Σ for either of the two standalone problems (2a) or (2b), as that would result in an overdetermination. In other words, when solving (2a) one cannot specify both u_1 and $\frac{\partial u_1}{\partial \mathbf{n_1}}$ on Σ, and likewise for (2b).

To avoid the overdetermination and still allow for separate solution of individual subproblems, P.L. Lions proposed to use one Robin boundary condition [20], formed as a linear combination of the two continuity conditions (2c). For any pair of constants $(p_1, p_2) \in \mathbb{R}^2$, this *transmission condition* yields the following combined formulation in lieu of (2):

$$\begin{cases} \Delta u_1 = f & \text{in } \Omega_1 \\ u_1 = 0 & \text{on } \overline{\Omega}_1 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathrm{n}_1} + p_1\right) u_1 = \left(\frac{\partial}{\partial \mathrm{n}_1} + p_1\right) u_2 & \text{on } \Sigma \end{cases} \tag{3a}$$

$$\begin{cases} \Delta u_2 = f \quad \text{in } \Omega_2 \\ u_2 = 0 \quad \text{on } \overline{\Omega}_2 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_2}} + p_2\right) u_2 = \left(\frac{\partial}{\partial \mathbf{n_2}} + p_2\right) u_1 \quad \text{on } \Sigma \end{cases} \tag{3b}$$

Each of the two subproblems (3) is individually well-defined in the sense that the third equation in either (3a) or (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 (3) to the original BVP (1) requires an additional analysis.

Lions conducted this analysis in [20]. He replaced the combined formulation (3) with the iteration:

$$\begin{cases} \Delta u_1^{(n+1)} = f \quad \text{in } \Omega_1 \\ u_1^{(n+1)} = 0 \quad \text{on } \overline{\Omega}_1 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_1}} + p_1\right) u_1^{(n+1)} = \left(\frac{\partial}{\partial \mathbf{n_1}} + p_1\right) u_2^{(n)} \quad \text{on } \Sigma \end{cases} \tag{4}$$

$$\begin{cases} \Delta u_2^{(n+1)} = f \quad \text{in } \Omega_2 \\ u_2^{(n+1)} = 0 \quad \text{on } \overline{\Omega}_2 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_2}} + p_2\right) u_2^{(n+1)} = \left(\frac{\partial}{\partial \mathbf{n_2}} + p_2\right) u_1^{(n)} \quad \text{on } \Sigma \end{cases} \tag{5}$$

and proved that this iteration converges to the solution of (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 [20] extends to an arbitrary number of subdomains.

2.2. Helmholtz adaptation

Complications arise when applying (5) directly to the Helmholtz equation. Consider the following BVP over the domain from Fig. 2a:

$$\begin{cases} \Delta u + k^2 u = f \quad \text{in } \Omega \\ u = 0 \qquad\qquad \text{on } \partial\Omega \end{cases} \tag{6}$$

To guarantee well-posedness of (6), i.e., to avoid resonances, $-k^2$ may not be an eigenvalue of the underlying Laplace problem. However, when considering a decomposition such as the one in Fig. 2b 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. For more elaborate decompositions, establishing the well-posedness of individual subproblems may only become more problematic. This issue was addressed in [12] where 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. (3)):

$$\begin{cases} (\Delta + k^2) u_1 = f \quad \text{in } \Omega_1 \\ u_1 = 0 \quad \text{on } \overline{\Omega}_1 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_1}} + ik\right) u_1 = \left(\frac{\partial}{\partial \mathbf{n_1}} + ik\right) u_2 \quad \text{on } \Sigma \end{cases} \tag{7a}$$

$$\begin{cases} (\Delta + k^2) u_2 = f \quad \text{in } \Omega_2 \\ u_2 = 0 \quad \text{on } \overline{\Omega}_2 \cap \partial\Omega \\ \left(\frac{\partial}{\partial \mathbf{n_2}} + ik\right) u_2 = \left(\frac{\partial}{\partial \mathbf{n_2}} + ik\right) u_1 \quad \text{on } \Sigma \end{cases} \tag{7b}$$

Després' transmission condition shifts the spectrum of the operator to the complex domain, guaranteeing that resonant frequencies are avoided on each subproblem (7a) or (7b). 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 (5) can be employed for (7), and Després showed in [12] that it will converge.

2.3. Cross-points and material discontinuities

The methods outlined in Section 2.2 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" [5]. However, while this leads to convergence in fewer iterations, it generally requires more expensive iterations.

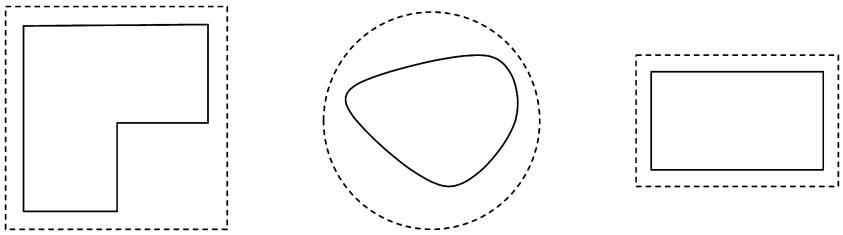


Fig. 3. Example domains (solid boundary) and a reasonable choice of auxiliary domain (dotted boundary) for each.

Work has also been done recently on 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 of DDMs if no special care is exercised for the discretization. In [16], several methods are discussed for resolving these cross-points for elliptic problems, and [25] provides an extension of the quasi-optimal method from [5] that accounts for interior cross-points. In Section 4, we demonstrate how the issue of cross-points is resolved intrinsically by our method, with no special consideration. Unlike most other methods (see, e.g., [9,10,18]), our method enforces the interface conditions exactly and thus is completely insensitive to the interior cross-points by design.

Material discontinuities and the associated transmission/scattering problems provide a natural venue for the application of DDMs, but they require special care in the high-contrast, high-frequency regime (see, e.g., [4]). Similar to the case of cross-points, our method is insensitive to large jumps in the wavenumber, as discussed further in Section 4.2.

3. Method of difference potentials

To introduce the Method of Difference Potentials [31], consider the inhomogeneous Helmholtz equation with a general (constant-coefficient) Robin boundary condition

$$\Delta u + k^2 u = f \tag{8a}$$

$$\alpha u + \beta \frac{\partial u}{\partial \mathbf{n}} = \phi \tag{8b}$$

over the domain $\Omega \subset \mathbb{R}^2$ depicted in Fig. 2a, as well as its decomposition depicted in Fig. 2b. In a similar manner to traditional DDMs, we split the problem into two separate subdomains as in (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 (8a), on the domain with an equivalent operator equation at the boundary (Calderon's boundary equation with projection [8,34]). Unlike the case of the classical boundary integral equations (BIEs), the reduction from the domain to the boundary with the help of Calderon's operators is universal. It is not tied to any specific boundary conditions and also not limited to constant coefficients. Also unlike in BIEs, Calderon's operators work with actual physical sources as opposed to artificial densities, such as that of a single- or double-layer potential. The well-posedness of Calderon's boundary formulation is inherited automatically from that of the volumetric boundary-value problem. The Calderon's BEP 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 individual subdomains Ω_1 and Ω_2 (see Fig. 2b). Then, 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 (8b), which is also formulated in terms of the Cauchy data of the solution. Altogether, the MDP solves a coupled problem on the combined boundary of all subdomains. Nonetheless, the Calderon operators for individual subdomains are computed independently and in parallel.

Indeed, the operators are computed with the help of an 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 Fig. 3. 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 Fig. 4, and solve the resulting auxiliary problems independently. When the subdomains have the same shape and the wavenumber is also the same, we can reuse the computed operators. The choice of an auxiliary domain, as well as details of how to efficiently account for identical subdomains, is discussed in Section 3.2.1.

In this section, we introduce the MDP and apply it in the framework of DDM. For a detailed account of the theory and derivation of the MDP, see [31]. For additional details on handling more complicated boundary conditions, general geometries, and variable coefficients, see [6,7,23,24].

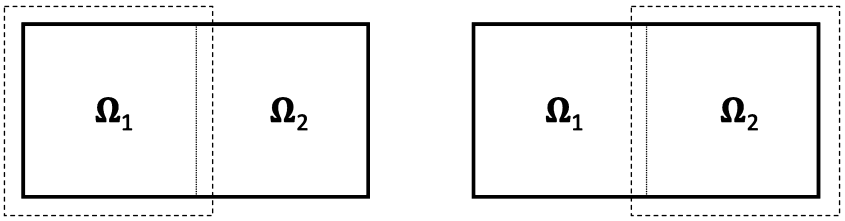


Fig. 4. The auxiliary domain setup for our problem with the domain decomposition from Fig. 2b.

Here is a roadmap for the rest of the section.

- Fourth order accurate finite difference scheme for the Helmholtz equation (Section 3.1).
- MDP for a single domain with no decomposition (Section 3.2).
  - Auxiliary problem for computing Calderon's potentials and boundary projections (Section 3.2.1).
  - Definition of difference potentials and boundary projection operators and equivalence of the finite difference Helmholtz equation to a discrete boundary equation with projection (Section 3.2.2).
  - Extension operator that relates the Cauchy data of the continuous solution at the boundary and the density of the difference potential on the grid (Section 3.2.3).
  - Representation of the Cauchy data at the boundary as a truncated spectral (Chebyshev) expansion with undetermined coefficients (Section 3.2.4).
  - Construction of a linear system with respect to the Chebyshev undetermined coefficients by substituting the spectral expansion into the boundary equation with projection via the extension operator (Section 3.2.5).
- MDP-based decomposition in the case of two subdomains (Section 3.3).
  - Combined system of two boundary equations with projection for two subdomains (Section 3.3.1).
  - Taking into account the boundary conditions using the previously introduced spectral representation of the data at the boundary (Section 3.3.2).
  - Taking into account the interface conditions in the spectral form by identifying and equating the respective Chebyshev expansion coefficients on the two sides of the interface (Section 3.3.3).
  - Solution of the resulting overdetermined linear system with respect to the unknown Chebyshev coefficients in the sense of least squares (Section 3.3.4).
- MDP-based decomposition in the case of more than two subdomains (Section 3.4).
- Summary of the algorithm (Section 3.5).
- Computational complexity of the algorithm (Section 3.6).

3.1. Finite difference scheme

The MDP can be implemented in conjunction with any finite difference scheme as the underlying approximation, including the case of complex or non-conforming boundaries [24]. High-order schemes are known to reduce the pollution effect for the Helmholtz equation [1,2,11]. Further, compact schemes require no additional boundary conditions beyond what is needed for the differential equation itself. Therefore, we have chosen to use the fourth-order, compact scheme for the Helmholtz equation as presented in [19,35]:

$$\begin{aligned}
&\frac{1}{h^2}\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}-4u_{m,n}\right)\\
&+\frac{1}{6h^2}\left[u_{m+1,n+1}+u_{m-1,n+1}+u_{m+1,n-1}+u_{m-1,n-1}+4u_{m,n}\right.\\
&\left.-2\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}\right)\right]\\
&+\frac{k^2}{12}\left(u_{m+1,n}+u_{m-1,n}+u_{m,n+1}+u_{m,n-1}+8u_{m,n}\right)\\
&=f_{m,n}+\frac{1}{12}\left(f_{m+1,n}+f_{m-1,n}+f_{m,n+1}+f_{m,n-1}-4f_{m,n}\right)
\end{aligned} \tag{9}$$

The scheme in (9) uses a nine-node stencil for the left-hand side of the PDE and a five-node stencil for the right-hand side (see Fig. 5), with uniform step size in both directions ($\Delta x = \Delta y = h$). In the case where the PDE is homogeneous, the right-hand side stencil is unnecessary as $f \equiv 0$. Additionally, (9) was derived for a constant value of the wavenumber

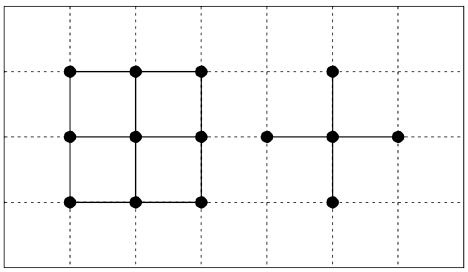


Fig. 5. The stencils for the compact scheme given in (9).

k. For our purposes this is sufficient, because while the domain Ω may have a piecewise-constant k, we assume that the decomposition is such that each Ω_i has constant k. One could also consider a sixth-order scheme for constant [39] or variable [36] wavenumber k, or a fourth-order scheme for a more general form of the Helmholtz equation with a variable coefficient Laplace-like term and wavenumber [7]. However, for the scope of this paper we will focus on the piecewise constant k case.

3.2. Base subdomain

For simplicity, we will first illustrate our approach for the case of identical subdomains. In this case, we can define all the components of the MDP on one base subdomain, and allow copies of that base subdomain to be translated and rotated into the appropriate position for any given concrete example. Considering the model domain from Fig. 2, a logical choice of base subdomain is a square. Throughout Section 3.2, we will refer to the base subdomain simply as Ω with boundary Γ, where Ω is a 2×2 square centered at the origin.

Then, we will consider a more general setting where different subdomains can have different values of the wavenumber k, including large differences between neighboring subdomains that will allow us to demonstrate the performance of the method for strong material discontinuities. Treating the partitions into subdomains of different shape, including non-rectangular shapes, presents no substantial difficulty either. It will be the subject of a future work.

3.2.1. Auxiliary problem

The auxiliary problem (AP) is needed for computing the Calderon's potentials and projections. It defines an inverse to the discrete Helmholtz operator. This inverse can be thought of as generalization of the convolution with the fundamental solution in the classical potential theory. The AP is formulated on a larger auxiliary domain $\Omega_0 \supset \Omega$. It should be uniquely solvable and well-posed, and should admit an efficient numerical solution. Otherwise, the AP can be arbitrary [31], and our specific choice is made for the reason of convenience.

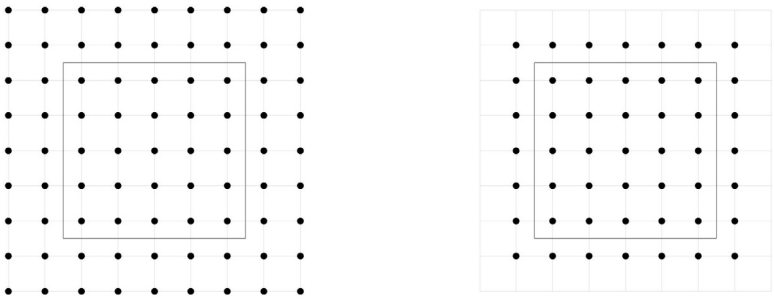
Let $\mathcal{L}$ be the Helmholtz operator: $\mathcal{L}u \stackrel{\text{def}}{=} (\Delta + k^2)u$, and Ω_0 be a square with side length 2.2. We formulate the AP by supplementing the inhomogeneous Helmholtz equation on Ω_0 with homogeneous Dirichlet conditions on the y-boundaries and local Sommerfeld conditions on the x-boundaries:

$$\begin{cases} \mathcal{L}u = g, & (x, y) \in \Omega_0 \\ u = 0, & y = \pm 1.1 \\ \frac{\partial u}{\partial x} + iku = 0, & x = 1.1 \\ \frac{\partial u}{\partial x} - iku = 0, & x = -1.1 \end{cases} \tag{10}$$

The choice of Sommerfeld-type conditions on the x-boundaries makes the spectrum of the AP (10) complex, guaranteeing that there is no resonance for any real wavenumber k. Hence, the AP (10) 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 2.2, the Sommerfeld-type conditions in (10) do not serve any transmission-related purpose, as they exist solely on the auxiliary domain and not on the physical boundary $\Gamma = \partial\Omega$.

To discretize the AP (10), we first replace the operator $\mathcal{L}$ with the left-hand side of the scheme (9):

$$\begin{aligned} &\frac{1}{h^2}\left(u_{m+1,n} + u_{m-1,n} + u_{m,n+1} + u_{m,n-1} - 4u_{m,n}\right) \\ &+ \frac{1}{6h^2}\left[u_{m+1,n+1} + u_{m-1,n+1} + u_{m+1,n-1} + u_{m-1,n-1} + 4u_{m,n}\right. \\ &\left. - 2\left(u_{m+1,n} + u_{m-1,n} + u_{m,n+1} + u_{m,n-1}\right)\right] \\ &+ \frac{k^2}{12}\left(u_{m+1,n} + u_{m-1,n} + u_{m,n+1} + u_{m,n-1} + 8u_{m,n}\right) = g_{m,n} \end{aligned} \tag{11a}$$



(a) $\mathbb{N}_0$ (b) $\mathbb{M}_0$

Fig. 6. Cartesian grid sets $\mathbb{N}_0$ and $\mathbb{M}_0$ for the stencils from Fig. 5. The larger overall square represents Ω_0; the smaller square in the middle with solid boundary is Ω with $\partial\Omega = \Gamma$.

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 $m = 0, \ldots, M$ set

$$u_{m,0} = u_{m,N} = 0 \tag{11b}$$

The following discretization of the Sommerfeld-type conditions was derived for the variable coefficient Helmholtz equation in [7] and simplified for the constant coefficient case in [6]:

$$\left(\frac{u_{M,n} - u_{M-1,n}}{h} - \frac{1}{6h}\left(u_{M,n+1} - u_{M-1,n+1} + u_{M,n-1} - u_{M-1,n-1} - 2\left(u_{M,n} - u_{M-1,n}\right)\right) - \frac{k^2 h}{24}\left(u_{M,n} - u_{M-1,n}\right)\right)$$
$$+ ik\left(\frac{u_{M,n} - u_{M-1,n}}{h} + \frac{h^2k^2}{8}u_{M-\frac{1}{2},n} + \frac{u_{M-\frac{1}{2},n+1} - 2u_{M-\frac{1}{2},n} + u_{M-\frac{1}{2},n-1}}{2}\right) = 0 \tag{11c}$$

$$\left(\frac{u_{1,n} - u_{0,n}}{h} - \frac{1}{6h}\left(u_{1,n+1} - u_{0,n+1} + u_{1,n-1} - u_{0,n-1} - 2\left(u_{1,n} - u_{0,n}\right)\right) - \frac{k^2 h}{24}\left(u_{1,n} - u_{0,n}\right)\right)$$
$$- ik\left(\frac{u_{1,n} - u_{0,n}}{h} + \frac{h^2k^2}{8}u_{\frac{1}{2},n} + \frac{u_{\frac{1}{2},n+1} - 2u_{\frac{1}{2},n} + u_{\frac{1}{2},n-1}}{2}\right) = 0 \tag{11d}$$

Conditions (11c) and (11d) were derived under the assumption that the right-hand side is compactly supported, i.e., the governing equation is homogeneous near the boundary. In our current setting, the grid function $g_{m,n}$ will be specified on the interior grid nodes, $m = 1, \ldots, M-1$ and $n = 1, \ldots, N-1$, and will be zero on the outermost grid nodes.

We define the discrete operator $\mathcal{L}^{(h)}$ as the application of the left-hand side of (11a), allowing the discrete AP to be expressed as $\mathcal{L}^{(h)}u = g$ subject to the boundary conditions from (11b), (11c), and (11d). The right-hand side g of the discrete AP can be arbitrary. Similar to the continuous AP (10), the finite difference AP (11) has a unique solution u for any $g = \{g_{m,n}\}$. This solution u defines the inverse operator $\mathcal{G}^{(h)}$: $u = \mathcal{G}^{(h)}g$.

In particular, the right-hand side g in (11a) may be defined as

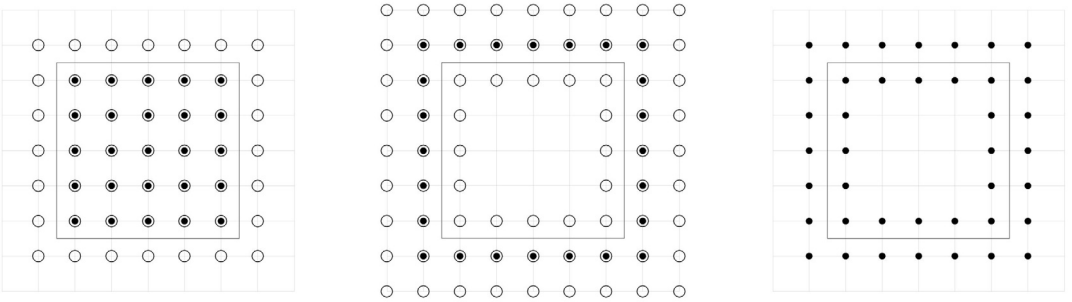
$$g_{m,n} = \mathcal{B}^{(h)} f_{m,n} \stackrel{\text{def}}{=} f_{m,n} + \frac{1}{12}\left(f_{m+1,n} + f_{m-1,n} + f_{m,n+1} + f_{m,n-1} - 4f_{m,n}\right) \tag{12}$$

where $\mathcal{B}^{(h)}$ represents the application of the stencil from the right-hand side of the scheme (9) to the source term f of the Helmholtz equation (8a). We emphasize that $\mathcal{G}^{(h)}$ is defined for any grid function g, not just those of the form $g = \mathcal{B}^{(h)}f$ as in (12). In Section 3.2.2, we introduce the discrete Calderon's potentials and projections as solutions to the AP obtained by applying $\mathcal{G}^{(h)}$ to specific source terms that are not in the form $\mathcal{B}^{(h)}f$ (see equation (13)).

The discrete AP (11) can be solved for any g by a combination of a sine-FFT in the y-direction and a tridiagonal solver in the x-direction. Note that, the use of the optimal log-linear solver presents no loss of generality as we have specifically formulated the discrete AP so that it would allow for an efficient numerical solution. The boundary conditions of the AP (11b), (11c), and (11d) that enable the application of this solver are set at the auxiliary boundary $\partial\Omega_0$ and impose no limitations for computing the Calderon's projections at the physical boundary $\Gamma = \partial\Omega$.

3.2.2. 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 Fig. 6). Define $\mathbb{M}^+ = \mathbb{M}_0 \cap \Omega$ as the nodes that are interior to



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

Fig. 7. Grid sets $\mathbb{M}^+$, $\mathbb{N}^+$, $\mathbb{M}^-$, $\mathbb{N}^-$, and γ. The larger overall square represents Ω_0; the smaller square in the middle with solid boundary is Ω with $\partial\Omega = \Gamma$.

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 Fig. 5(left) to every node in $\mathbb{M}^+$, and similarly let $\mathbb{N}^-$ be the same for $\mathbb{M}^-$ (see Figs. 7a and 7b). Finally, we define the grid boundary $\gamma = \mathbb{N}^+ \cap \mathbb{N}^-$ as the discrete analogue of the original problem's boundary $\Gamma = \partial\Omega$ (see Fig. 7c). Consider a grid function $\boldsymbol{\xi}_\gamma$ specified on the discrete boundary γ. We can then define the *difference potential with density* $\boldsymbol{\xi}_\gamma$ as

$$P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma \stackrel{\text{def}}{=} w - \mathcal{G}^{(h)}\left(\mathcal{L}^{(h)} w\big|_{\mathbb{M}^+}\right), \quad \text{where} \quad w = \begin{cases} \boldsymbol{\xi}_\gamma & \text{on } \gamma \\ 0 & \text{on } \mathbb{N}_0 \backslash \gamma \end{cases} \tag{13}$$

The operation $\mathcal{L}^{(h)} w\big|_{\mathbb{M}^+}$ in (13) represents first applying the operator $\mathcal{L}^{(h)}$ to the grid function w, then truncating the result to the grid set $\mathbb{M}^+$. The difference potential $P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma$ is a grid function defined on $\mathbb{N}^+$ (hence the notation). By design, it satisfies the homogeneous finite difference equation $\mathcal{L}^{(h)}(P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma) = 0$ on $\mathbb{M}^+$. The difference potential (13) truncated to the grid boundary γ defines the *discrete boundary projection operator* P_γ:

$$P_\gamma \boldsymbol{\xi}_\gamma \stackrel{\text{def}}{=} (P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma)\big|_\gamma \equiv Tr^{(h)}(P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma) \tag{14}$$

The projection P_γ given by (14) has the following key property:

Theorem 1 *(Ryaben'kii). A grid function $\boldsymbol{\xi}_\gamma$ satisfies the difference boundary equation with projection*

$$P_\gamma \boldsymbol{\xi}_\gamma + Tr^{(h)}\mathcal{G}^{(h)} g = \boldsymbol{\xi}_\gamma \tag{15}$$

if and only if there is a solution u on $\mathbb{N}^+$ to the finite difference equation (11a) such that $\boldsymbol{\xi}_\gamma$ is the trace (truncation) of u on the grid boundary γ, i.e., $\boldsymbol{\xi}_\gamma = Tr^{(h)} u$.

The operator $Tr^{(h)}$ in equation (15) is the same truncation to γ (or, equivalently, trace on γ) as introduced first in equation (14). The proof of Theorem 1 (as well as that of the projection property, $P_\gamma^2 = P_\gamma$) can be found in [31, Part II, Chapter 2]. If the BEP (15) holds on γ, then the solution u on $\mathbb{N}^+$ is reconstructed by means of the discrete generalized Green's formula:

$$u = P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma + \mathcal{G}^{(h)} g \tag{16}$$

In particular, the discrete right-hand side g in equations (15) and (16) may be given by (12): $g = \mathcal{B}^{(h)} f$. Then, the discrete BEP (15) equivalently reduces the fourth order accurate discrete approximation of the Helmholtz equation $\mathcal{L}u = f$ 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, the BEP (15) reduces to

$$P_\gamma \boldsymbol{\xi}_\gamma = \boldsymbol{\xi}_\gamma \tag{17}$$

and we have

Corollary 2. *A grid function $\boldsymbol{\xi}_\gamma$ satisfies the homogeneous difference BEP (17) if and only if there is a solution u on $\mathbb{N}^+$ to the finite difference equation $\mathcal{L}^{(h)} u = 0$ such that $\boldsymbol{\xi}_\gamma$ is the trace of u on the grid boundary γ, i.e., $\boldsymbol{\xi}_\gamma = Tr^{(h)} u$.*

Similar to (15) and (16), solutions of the BEP (17) can be used to reconstruct the corresponding solution u on $\mathbb{N}^+$ in the form of the difference potential (13):

$$u = P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma \tag{18}$$

3.2.3. Equation-based extension

In order for u from (16) to approximate the solution of (8a) on $\mathbb{N}^+$, the grid density $\boldsymbol{\xi}_\gamma$ 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: $\boldsymbol{\xi}_\Gamma = (\xi_0, \xi_1)\big|_\Gamma$. One can interpret ξ_0 and ξ_1 as the Dirichlet and Neumann data, respectively, of some function $v = v(x, y)$ on Ω_0:

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

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

$$v(x, y) \stackrel{\text{def}}{=} v|_\Gamma + \rho \left.\frac{\partial v}{\partial \mathbf{n}}\right|_\Gamma + \frac{\rho^2}{2}\left.\frac{\partial^2 v}{\partial \mathbf{n}^2}\right|_\Gamma + \frac{\rho^3}{6}\left.\frac{\partial^3 v}{\partial \mathbf{n}^3}\right|_\Gamma + \frac{\rho^4}{24}\left.\frac{\partial^4 v}{\partial \mathbf{n}^4}\right|_\Gamma \tag{19}$$

The definition (19) 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 (8a), 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 $x = X = \text{const}$. Then, the outward normal derivative becomes the positive $x-$derivative, and by rearranging (8a), we immediately get an expression for the second x-derivative evaluated along Γ:

$$\frac{\partial^2 v}{\partial x^2}(X, y) = f(X, y) - \frac{\partial^2 v}{\partial y^2}(X, y) - k^2 v(X, y) \tag{20}$$

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 (8a) 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 (20). This process yields the following expressions:

$$v(X, y) = \xi_0(y) \tag{21a}$$

$$\frac{\partial v}{\partial x}(X, y) = \xi_1(y) \tag{21b}$$

$$\frac{\partial^2 v}{\partial x^2}(X, y) = f(X, y) - \xi_0''(y) - k^2\xi_0(y) \tag{21c}$$

$$\frac{\partial^3 v}{\partial x^3}(X, y) = \frac{\partial f}{\partial x}(X, y) - \xi_1''(y) - k^2\xi_1(y) \tag{21d}$$

$$\frac{\partial^4 v}{\partial x^4}(X, y) = \frac{\partial^2 f}{\partial x^2}(X, y) - \frac{\partial^2 f}{\partial y^2}(X, y) - k^2 f(X, y) + \xi_0^{(4)}(y) + 2k^2\xi_0^{(2)}(y) + k^4\xi_0(y) \tag{21e}$$

We reiterate that, all functions in formulae (21) are functions of a continuous argument (not grid functions) and accordingly, their derivatives are conventional derivatives rather than finite differences. The expressions in (21) can be substituted into (19) to calculate the values of $v(x, y)$ near the right side of Ω. 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 (21a)-(21e) into the Taylor expansion (19). Then, sampling v only on the grid boundary γ, we define the extension operator $\mathbf{Ex}$ that for a given pair of continuous functions (ξ_0, ξ_1) on Γ yields a discrete function $\boldsymbol{\xi}_\gamma$ on γ:

$$\boldsymbol{\xi}_\gamma = \mathbf{Ex}(\xi_0, \xi_1) \stackrel{\text{def}}{=} v\big|_\gamma$$

As seen in (21), the operator $\mathbf{Ex}$ depends on the source term f. Hence, $\mathbf{Ex}$ is an affine operator:

$$\mathbf{Ex}\boldsymbol{\xi}_\Gamma = \mathbf{Ex}^{(H)}(\xi_0, \xi_1) + \mathbf{Ex}^{(I)} f \tag{22}$$

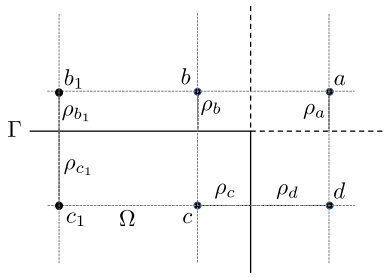


Fig. 8. Construction of the extension operator near the corner.

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 (8a).

Note that, for most nodes in γ, such as nodes b, b_1, c, c_1, and d in Fig. 8, the shortest distance to the continuous boundary Γ is defined unambiguously and the application of the Taylor formula (19) is straightforward. For nodes in exterior quadrants near the corners, such as node a in Fig. 8, the operator $\mathbf{Ex}$ is constructed as follows. The normal ρ_a from $a \in \gamma$ is dropped on the extension of a side of the square. The Taylor formula (19) remains the same as before, but the data at the foot of the normal ρ_a, which does not rest on the boundary Γ proper, are obtained by extending the function (ξ_0, ξ_1) outward along the corresponding segment of Γ. In the actual implementation of our algorithm, the functions on Γ that need to be extended are Chebyshev polynomials (see Sections 3.2.4 and 3.2.5). To obtain the data the foot of the normal ρ_a, a polynomial defined on a given side of the square is merely considered on an appropriate larger interval.

We also emphasize that, although the formulae for the normal derivatives (21) were derived using the Helmholtz equation, $\boldsymbol{\xi}_\Gamma = (\xi_0, \xi_1)$ does not need to represent the Cauchy data of a solution u to (8a) in order to apply the operator $\mathbf{Ex}$. However, if $\boldsymbol{\xi}_\Gamma$ does correspond to a solution u: $\boldsymbol{\xi}_\Gamma = \left(u, \frac{\partial u}{\partial \mathbf{n}}\right)\big|_\Gamma$, then $\boldsymbol{\xi}_\gamma = \mathbf{Ex}\boldsymbol{\xi}_\Gamma$ approximates this solution near Γ with fifth-order accuracy with respect to the grid size h, specifically at the grid nodes of γ.

Theorem 3 *(Reznik). Let u be a solution to equation (8a) on Ω in the homogeneous case, $f \equiv 0$, and let $\boldsymbol{\xi}_\Gamma$ be the trace of u along the continuous boundary Γ such that $\boldsymbol{\xi}_\Gamma = \left(u, \frac{\partial u}{\partial \mathbf{n}}\right)\big|_\Gamma$. Let $\boldsymbol{\xi}_\gamma = \mathbf{Ex}\boldsymbol{\xi}_\Gamma$ and let $P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma$ be the difference potential with density $\boldsymbol{\xi}_\gamma$ defined by formula (13). Let p be the order of accuracy of the finite difference scheme $\mathcal{L}^{(h)}u = 0$. Then, as the grid $\mathbb{N}_0$ is refined, the difference potential $P_{\mathbb{N}^+}\boldsymbol{\xi}_\gamma$ converges to the solution u (on the grid $\mathbb{N}^+$) with the rate of $\mathcal{O}(h^p)$ provided that the order of the Taylor's formula (19) is equal to $p + q$, where q is the order of the differential operator $\mathcal{L}$.*

The proof of Theorem 3 can be found in [29], as well as in [31, Part III, Section 1.4]. Since the Helmholtz equation is second-order, Theorem 3 suggests that the fourth-order Taylor's formula (19) can be used for constructing the extension operator in conjunction with a second-order accurate finite difference scheme $\mathcal{L}^{(h)}u = 0$. However, the scheme (9) that we are using is fourth-order accurate. Hence, according to Theorem 3, one may need a sixth-order Taylor's formula for the extension operator. Yet in practice, it has repeatedly been observed [6,23,24] that while sufficient, the bound given by Theorem 3 is not tight, and the order of the Taylor's formula can be taken equal to the order of the finite difference scheme alone. Our use of a fourth-order Taylor's formula (19) is corroborated by the numerical simulations in Section 4.

3.2.4. Spectral representation of the boundary data

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

$$\boldsymbol{\psi}_j^{(0)} = \left(\psi_j, 0\right), \quad \boldsymbol{\psi}_j^{(1)} = \left(0, \psi_j\right), \quad j = 1, ..., \infty \tag{23}$$

Recall that we denote the boundary data by $\boldsymbol{\xi}_\Gamma = (\xi_0, \xi_1)$, where ξ_0 represents the Dirichlet data and ξ_1 represents the Neumann data. Denote by Γ^* any smooth section of Γ (i.e., any one side of the square), and let its boundary data be $\boldsymbol{\xi}_{\Gamma^*} = \left(\xi_0^*, \xi_1^*\right)$. For all simulations in this paper, we take $\{\psi_j\}$ as Chebyshev polynomials defined on Γ^* and introduce the expansion:

$$\boldsymbol{\xi}_{\Gamma^*} = \left(\xi_0^*, \xi_1^*\right) = \sum_{j=1}^{\infty} c_j^{(0)} \boldsymbol{\psi}_j^{(0)} + \sum_{j=1}^{\infty} c_j^{(1)} \boldsymbol{\psi}_j^{(1)} \tag{24}$$

where $c_j^{(0)}$ and $c_j^{(1)}$, $j = 1, 2, \ldots$, are the respective Chebyshev coefficients. The infinite series (24) can be truncated after a finite number of terms to provide an approximation of $\boldsymbol{\xi}_{\Gamma^*}$. 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:

$$\boldsymbol{\xi}_{\Gamma^*} = \sum_{j=1}^{M^*} c_j^{(0)} \boldsymbol{\psi}_j^{(0)} + \sum_{j=1}^{M^*} c_j^{(1)} \boldsymbol{\psi}_j^{(1)} \tag{25}$$

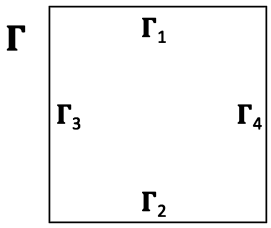


Fig. 9. Labeling of the sides of Γ (helps define the linear system in Section 3.2.5).

For smooth boundary data, the Chebyshev expansion (24) converges rapidly [32, Section 3.1.3] and thus the number of terms M^* in the truncated sum (25) can be small. The specific values of M^* that we used in our simulations are presented in Section 4.

The spectral representation (25) 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 Fig. 9, and the following definition of the expanded set of basis functions Ψ_j:

$$\Psi_{j+(i-1)M^*} = \begin{cases} \psi_j & \text{on } \Gamma_i \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, \ldots, 4 \tag{26}$$

Every element of Ψ in (26) is defined on the entire Γ, while each ξ_{Γ^i} has a series expansion independent of the others. Then, similar to (23) we define the following pairs:

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

where $M = 4 \cdot M^*$, and write the expansion of $\boldsymbol{\xi}_\Gamma$ as

$$\boldsymbol{\xi}_\Gamma = \sum_{j=1}^{M} c_j^{(0)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} c_j^{(1)} \boldsymbol{\Psi}_j^{(1)} \tag{27}$$

Note that, the choice of the same system of basis functions for both the Dirichlet and Neumann data and for all four sides of the square is not a requirement, but it provides extra convenience for constructing the linear system in Section 3.2.5 and building the DDM extension in Section 3.3.

3.2.5. Linear system for one subdomain

Applying the extension operator (22) to the series representation of $\boldsymbol{\xi}_\Gamma$ in (27), we have:

$$\begin{aligned} \mathbf{E}\mathbf{x}\boldsymbol{\xi}_\Gamma &= \mathbf{E}\mathbf{x}^{(H)} \left(\sum_{j=1}^{M} c_j^{(0)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} c_j^{(1)} \boldsymbol{\Psi}_j^{(1)} \right) + \mathbf{E}\mathbf{x}^{(I)} f \\ &= \sum_{j=1}^{M} c_j^{(0)} \mathbf{E}\mathbf{x}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} c_j^{(1)} \mathbf{E}\mathbf{x}^{(H)} \boldsymbol{\Psi}_j^{(1)} + \mathbf{E}\mathbf{x}^{(I)} f \end{aligned} \tag{28}$$

Setting $\boldsymbol{\xi}_\gamma = \mathbf{E}\mathbf{x}\boldsymbol{\xi}_\Gamma$ and substituting it into the BEP (15) with $g = \mathcal{B}^{(h)} f$ yields:

$$\begin{aligned} & P_\gamma \boldsymbol{\xi}_\gamma = \boldsymbol{\xi}_\gamma - Tr^{(h)} \mathcal{G}^{(h)} \mathcal{B}^{(h)} f \\ & P_\gamma \mathbf{E}\mathbf{x}\boldsymbol{\xi}_\Gamma = \mathbf{E}\mathbf{x}\boldsymbol{\xi}_\Gamma - Tr^{(h)} \mathcal{G}^{(h)} \mathcal{B}^{(h)} f \\ & P_\gamma \left(\sum_{j=1}^{M} c_j^{(0)} \mathbf{E}\mathbf{x}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} c_j^{(1)} \mathbf{E}\mathbf{x}^{(H)} \boldsymbol{\Psi}_j^{(1)} + \mathbf{E}\mathbf{x}^{(I)} f \right) \\ & \quad = \sum_{j=1}^{M} c_j^{(0)} \mathbf{E}\mathbf{x}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} c_j^{(1)} \mathbf{E}\mathbf{x}^{(H)} \boldsymbol{\Psi}_j^{(1)} + \mathbf{E}\mathbf{x}^{(I)} f - Tr^{(h)} \mathcal{G}^{(h)} \mathcal{B}^{(h)} f \\ & \sum_{j=1}^{M} c_j^{(0)} P_\gamma \mathbf{E}\mathbf{x}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} c_j^{(1)} P_\gamma \mathbf{E}\mathbf{x}^{(H)} \boldsymbol{\Psi}_j^{(1)} + P_\gamma \mathbf{E}\mathbf{x}^{(I)} f \end{aligned}$$

$$= \sum_{j=1}^{M} c_j^{(0)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} c_j^{(1)} \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(1)} + \mathbf{Ex}^{(I)} f - Tr^{(h)} \mathcal{G}^{(h)} \mathcal{B}^{(h)} f$$

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

$$\begin{aligned} &\sum_{j=1}^{M} c_j^{(0)} (P_\gamma - I_\gamma) \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(0)} + \sum_{j=1}^{M} c_j^{(1)} (P_\gamma - I_\gamma) \mathbf{Ex}^{(H)} \boldsymbol{\Psi}_j^{(1)} \\ &\quad = (I_\gamma - P_\gamma) \mathbf{Ex}^{(I)} f - Tr^{(h)} \mathcal{G}^{(h)} \mathcal{B}^{(h)} f \end{aligned} \tag{29}$$

where I_γ represents the identity operator in the space of grid functions $\boldsymbol{\xi}_\gamma$ defined on γ. While equations (29) hold at the nodes of the grid boundary γ, the unknowns in system (29) are not grid unknowns; they are rather the coefficients $c_j^{(0)}$ and $c_j^{(1)}$, $j = 1, 2, \ldots, M$, of the Chebyshev expansion on Γ of the Dirichlet and Neumann data of the continuous solution. System (29) can be written in matrix form:

$$Q\, c = F \tag{30}$$

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

$$Q = \Big[\underbrace{(P_\gamma - I_\gamma) \mathbf{Ex}\, \boldsymbol{\Psi}_1^{(0)},\ \ldots,\ (P_\gamma - I_\gamma) \mathbf{Ex}\, \boldsymbol{\Psi}_M^{(0)}}_{Q^{(0)}}, \underbrace{(P_\gamma - I_\gamma) \mathbf{Ex}\, \boldsymbol{\Psi}_1^{(1)},\ \ldots,\ (P_\gamma - I_\gamma) \mathbf{Ex}\, \boldsymbol{\Psi}_M^{(1)}}_{Q^{(1)}} \Big] \tag{31}$$

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

$$\begin{aligned} c &= [c_1^{(0)},\ \ldots,\ c_M^{(0)},\ c_1^{(1)},\ \ldots,\ c_M^{(1)}] \\ &\equiv [\underbrace{c_1,\ \ldots,\ c_M}_{c^{(0)\mathsf{T}}},\ \underbrace{c_{M+1},\ \ldots,\ c_{2M}}_{c^{(1)\mathsf{T}}}]^{\mathsf{T}} \end{aligned} \tag{32}$$

in equation (30) is a vector of unknowns with dimension $2M$, while the vector F has dimension $|\gamma|$ and represents the inhomogeneous part of the problem:

$$F = (I_\gamma - P_\gamma) \mathbf{Ex}^{(I)} f - Tr^{(h)} \mathcal{G}^{(h)} \mathcal{B}^{(h)} f \tag{33}$$

The first M columns of Q in (31) form the sub-matrix $Q^{(0)}$ and correspond to the coefficients $\{c_j^{(0)}\}$ in (32), while columns $M+1$ through $2M$ form $Q^{(1)}$ and correspond to $\{c_j^{(1)}\}$.

Note that, the solution to (30) is not unique, as system (30) is derived from the discretized governing differential equation $\mathcal{L}^{(h)} u = \mathcal{B}^{(h)} f$ with the help of the BEP (15), and does not take into account any boundary conditions. Therefore, we interpret system (30) as a core component of the multi-subdomain decomposition algorithm, rather than a system to be solved in its own right. The decomposition algorithm is described in Section 3.3 for the case of two subdomains and subsequently extended in Section 3.4 to the case of a larger number of subdomains. For a discussion about implementing boundary conditions and completing the MDP algorithm in the single domain case, see [6,7,23,24,31].

3.3. Extension to two subdomains

We reconsider the problem of solving (8) over a partitioned domain as in Fig. 2b.

3.3.1. Combined system of two BEPs

Let $\Gamma^{(i)}$ represent the boundary of Ω_i, and let each $\Gamma^{(i)}$ be composed of its four sides as in Fig. 9, 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. According to Fig. 9, 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 3.2 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:

$$\begin{aligned} Q^{(i,*,l)} &= \left[Q^{(i,1,l)} \quad Q^{(i,2,l)} \quad Q^{(i,3,l)} \quad Q^{(i,4,l)} \right] \\ c^{(i,*,l)} &= \left[c^{(i,1,l)\mathsf{T}} \quad c^{(i,2,l)\mathsf{T}} \quad c^{(i,3,l)\mathsf{T}} \quad c^{(i,4,l)\mathsf{T}} \right]^{\mathsf{T}} \end{aligned}$$

For $Q^{(i,j,l)}$, 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 $l \in \{0,1\}$ distinguishes between the Dirichlet and Neumann data (compare to the notation

$Q^{(0)}$ and $Q^{(1)}$ in Section 3.2.5). The use of $c^{(i,j,l)}$ 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

$$Q^{(1,*,*)}c^{(1,*,*)} = F^{(1)} \quad \text{and} \quad Q^{(2,*,*)}c^{(2,*,*)} = F^{(2)} \tag{34}$$

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 $Q^{(i,*,*)}$, $i \in \{1,2\}$, is identical to that of (31) over a single subdomain. Equivalently, the independent linear systems (34) can be expressed simultaneously as the block-diagonal system

$$\begin{bmatrix} Q^{(1,*,*)} & 0 \\ 0 & Q^{(2,*,*)} \end{bmatrix} \begin{bmatrix} c^{(1,*,*)} \\ c^{(2,*,*)} \end{bmatrix} = \begin{bmatrix} F^{(1)} \\ F^{(2)} \end{bmatrix} \tag{35}$$

Similar to (30), the solution to (35) is not unique because it is derived only from the discrete BEP (15) combined with (12) for each subdomain and does not account for the boundary condition (8b). Additionally, since Ω has been decomposed into Ω_1 and Ω_2, an interface condition is needed to account for the lack of a boundary condition along Σ.

Note that, while the boundary value problem (8) on the composite domain shown in Fig. 2b assumes the same value of k throughout the entire Ω, it is very easy to generalize equations (34) to the case where Ω_1 and Ω_2 have different wavenumbers. That would require computing $Q^{(1,*,*)}$ for the wavenumber k_1 and computing $Q^{(2,*,*)}$ for the wavenumber k_2, which is done by solving two similar yet different discrete APs (11).

3.3.2. 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 (25) into the boundary condition (8b) for both u and $\frac{\partial u}{\partial \mathbf{n}}$. Expand the right-hand side of (8b) as $\phi = \sum_m^{M^*} d_m^{(i,j)} \psi_m$ using the same basis functions as in (23). Then,

$$\alpha \left(\sum_{m=1}^{M^*} c_m^{(i,j,0)} \psi_m \right) + \beta \left(\sum_{m=1}^{M^*} c_m^{(i,j,1)} \psi_m \right) = \sum_{m}^{M^*} d_m^{(i,j)} \psi_m \tag{36}$$

Assuming that the basis functions ψ_m are orthogonal, we derive from (36):

$$\alpha c_m^{(i,j,0)} + \beta c_m^{(i,j,1)} = d_m^{(i,j)}, \quad \text{for} \quad m \in \{1, \ldots, M^*\} \tag{37}$$

The M^* equations (37) 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 (8b). For example, the definitions of α and β in (8b) can be piece-wise constant:

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

where $\alpha^{(i,j)}$ and $\beta^{(i,j)}$ are constants that can assume different values on different segments of the boundary $\Gamma^{(i,j)}$ as long as $\left(\alpha^{(i,j)}, \beta^{(i,j)}\right) \neq (0,0)$ for any pair $(i,j) \in B$. This generalization allows for both Dirichlet ($\alpha^{(i,j)} = 1$, $\beta^{(i,j)} = 0$) and Neumann ($\alpha^{(i,j)} = 0$, $\beta^{(i,j)} = 1$) conditions as particular cases. The equations being added by (37) are sparse compared to the rest of (35), which can be taken advantage of computationally, see Section 3.3.4. Further information on implementing mixed boundary conditions, as well as extending this process to include variable coefficient Robin conditions, can be found in [6].

3.3.3. Interface conditions

The standard interface conditions require continuity of the solution and its flux across the interface (see equations (2c)). 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 $\Gamma^{(2,3)}$:

$$\sum_{m=1}^{M^*} c_m^{(1,4,0)} \psi_m = \sum_{m=1}^{M^*} c_m^{(2,3,0)} \psi_m \tag{38a}$$

$$\sum_{m=1}^{M^*} c_m^{(1,4,1)} \psi_m = -\sum_{m=1}^{M^*} c_m^{(2,3,1)} \psi_m \tag{38b}$$

As with the boundary conditions in Section 3.3.2, the use of identical systems of Chebyshev polynomials $\{\psi_j\}$ (orthogonal basis functions) along each side is exploited to obtain the following two sets of equations for each $m \in \{1, \ldots, M^*\}$:

$$c_m^{(1,4,0)} - c_m^{(2,3,0)} = 0 \tag{39a}$$

$$c_m^{(1,4,1)} + c_m^{(2,3,1)} = 0 \tag{39b}$$

The $2M^*$ additional equations (39) supplement system (35) along with boundary conditions (8b) expressed in the form (37).

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 $\Gamma^{(1,4)} = \Gamma^{(2,3)}$:

$$a^{(0)}u_1 + b^{(0)}u_2 = \eta^{(0)} \qquad a^{(1)}\frac{\partial u_1}{\partial \mathbf{n_1}} + b^{(1)}\frac{\partial u_2}{\partial \mathbf{n_2}} = \eta^{(1)} \tag{40}$$

The basic interface conditions (2c) correspond to $a^{(0)} = -b^{(0)} = 1$, $a^{(1)} = b^{(1)} = 1$, and $\eta^{(0)} = \eta^{(1)} \equiv 0$ in (40). Any interface conditions (40) can be accounted for by following the same steps as in (38) and (39). 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

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

and

$$a^{(1)}c_m^{(1,4,1)} + b^{(1)}c_m^{(2,3,1)} = \eta_m^{(1)}$$

as the conditions for the coefficients. 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 paper, for simplicity, we only consider the case where the solution and its flux are continuous on Σ.

3.3.4. Solving the overall system

By supplementing the system (35) with the equations derived in (37) and (39), the complete system can be expressed as a new matrix equation

$$\overline{Q}\, c^{(*,*,*)} = \overline{F} \tag{41}$$

where the dimension of $\overline{Q}$ is $(2|\gamma| + 8M^*) \times 16M^*$. Again, we emphasize that while the number of equations in (41) corresponds to the number of nodes in the grid boundary, the unknowns $c^{(*,*,*)}$ in system (41) are not grid unknowns. They are the coefficients of the spectral expansion (Chebyshev) of the continuous solution and its normal derivative at the boundaries of all subdomains. The system (41) is overdetermined and 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. The least squares solution exists and is unique. Moreover, as the grid is refined the ℓ_2 residual of system (41) converges to zero with the design rate of the scheme $\mathcal{O}(h^4)$. This convergence takes place because although (41) is solved weakly, the original boundary value problem (8), from which (41) is ultimately derived, has a classical solution.

Rather than adding equations (37) and (39) to the system, these conditions can instead be resolved through substitution and the elimination of unknowns. For the boundary conditions in the form (37), first consider the case where (8b) 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 (8b), 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 (35). If (8b) reduces to a Neumann boundary 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 (8b) 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 (37), and rearrange the terms to solve for either $c_m^{(i,j,0)}$ or $c_m^{(i,j,1)}$:

$$c_m^{(i,j,1)} = \frac{1}{\beta}d_m^{(i,j)} - \frac{\alpha}{\beta}c_m^{(i,j,0)} \tag{42}$$

From (42), 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 (35). Similar to the Dirichlet and Neumann cases, $6M^*$ unknowns are eliminated from the system.

The interface conditions (39a) 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 (39b) (subtracting columns instead of adding) eliminates an additional M^* unknowns.

In the case where (37) and (39) are included as supplemental equations, the overall system has dimension $(2|\gamma| + 8M^*) \times 16M^*$. If the conditions are resolved, the dimension is $2|\gamma| \times 8M^*$, which enables faster solution. The solution vector $c^{(*,*,*)}$ is used to reconstruct $\xi_{\Gamma^{(1)}}$ and $\xi_{\Gamma^{(2)}}$ through the series representation (25) for each subproblem. In turn, $\xi_{\Gamma^{(1)}}$ and $\xi_{\Gamma^{(2)}}$ are extended to their respective grid boundaries, as described in Section 3.2.3. Finally, a fourth-order accurate approximation to the unique solution of (8) is obtained by applying (16) to the resulting $\xi_{\gamma^{(1)}} = \mathbf{Ex}\xi_{\Gamma^{(1)}}$ and $\xi_{\gamma^{(2)}} = \mathbf{Ex}\xi_{\Gamma^{(2)}}$. These approximations collectively provide an approximation of the global solution to (8) on the overall domain Ω.

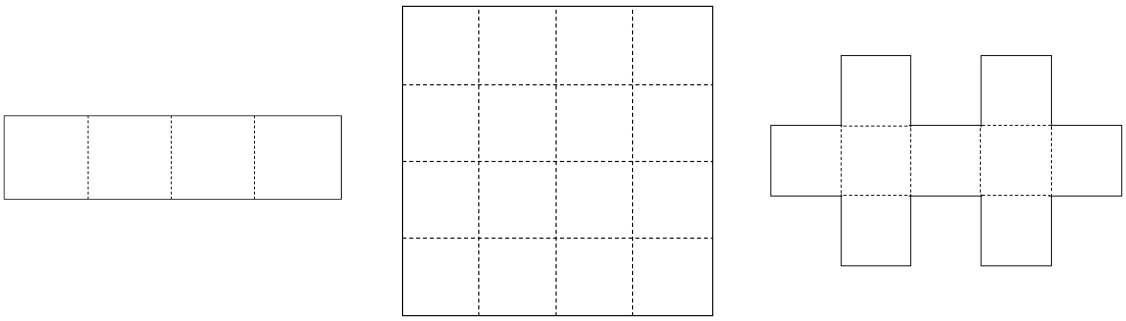


Fig. 10. Examples of domains that have a valid N-subdomain decomposition.

3.4. Extension to N subdomains

The extension to N subdomains is a natural extension of the two-subdomain case. Consider (8) over a domain Ω that is split into N subdomains, whose interfaces are full edges of the squares (see Fig. 10). Returning to the triple index notation used in Section 3.3, let the first argument vary from 1 to N, rather than stopping at 2, and let all grid sets and operators from Section 3.2 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 right-hand sides from each subdomain are simply appended to create the following system:

$$\begin{bmatrix} Q^{(1,*,*)} & 0 & 0 & 0 \\ 0 & Q^{(2,*,*)} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Q^{(N,*,*)} \end{bmatrix} \begin{bmatrix} c^{(1,*,*)} \\ c^{(2,*,*)} \\ \vdots \\ c^{(N,*,*)} \end{bmatrix} = \begin{bmatrix} F^{(1)} \\ F^{(2)} \\ \vdots \\ F^{(N)} \end{bmatrix} \tag{43}$$

The matrices $Q^{(i,*,*)}$ on the diagonal of (43) may correspond to the same wavenumber or different wavenumbers. In the latter case, computing $Q^{(i,*,*)}$ for a new wavenumber requires a new discrete AP (11). To generalize the handling of boundary and interface conditions, extend the definition of the set B

$$B = \left\{ (i,\, j) \middle| \Gamma^{(i,j)} \cap \partial\Omega \neq \emptyset \right\} \tag{44}$$

so that $|B \cup B^{\complement}| = 4N$. If $(i,\, j) \in B$, then $\Gamma^{(i,j)}$ has an associated boundary condition specified by (8b) and the process described in Section 3.3.2 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 3.3.3 to determine the supplemental equations. Adding these equations yields the N subdomain version of (41):

$$\overline{Q}_N c^{(*,*,*)} = \overline{F}_N \tag{45}$$

where $\overline{Q}_N$ and $\overline{F}_N$ represent the matrix from the left-hand side of (43) and the vector of the right-hand side, respectively, after being supplemented with boundary and interface condition equations. Similar to (41), the unknowns $c^{(*,*,*)}$ in system (45) are not grid unknowns; they are the coefficients of the Chebyshev expansion of the continuous solution at the boundary. The structure of the decomposition (see examples in Fig. 10) determines how many equations correspond to boundary conditions as opposed to interface conditions, but there will always be $4NM^*$ equations added to (43) (M^* equations for each $\Gamma^{(i,j)}$). As in Section 3.3.4, these equations can often be resolved with substitution and elimination to reduce the cost of solving the linear system.

3.5. Implementation details

In this section, we provide the important implementation details of the proposed algorithm, which are further justified in Section 3.6. We assume that all the subdomains are squares of the same size, as in Fig. 10. We allow different values of k on different subdomains so that in (8a) we assume that the wavenumber k is piece-wise constant over Ω, and constant on any given Ω_i. Consider the following summary of the algorithm:

1. For $i = 1, 2, \ldots, N$, consider the subdomain Ω_i:
   (a) Define the auxiliary problem (10), as well as the grid sets and operators from Section 3.2.2.
   (b) For $j = 1, 2, \ldots, M$:
   Compute $(P_\gamma - I_\gamma)\mathbf{Ex}\,\boldsymbol{\Psi}_j^{(0)}$ and $(P_\gamma - I_\gamma)\mathbf{Ex}\,\boldsymbol{\Psi}_j^{(1)}$.

(c) Compose the matrix $Q^{(i,*,*)}$ according to (31). $Q^{(i,*,*)}$ is the left-hand side of (30).
(d) Compute $P_\gamma \mathbf{Ex}^{(I)} f$ and $Tr^{(h)} \mathcal{G}^{(h)} \mathcal{B}^{(h)} f$ to form $F^{(i)}$ according to (33). $F^{(i)}$ is the right-hand side of (30).

2. Solve the system (45) in the sense of least squares and obtain $c^{(*,*,*)}$:

(a) Assemble the system (43) using the matrices $Q^{(i,*,*)}$ and vectors $F^{(i)}$ from step 1.
(b) Resolve the boundary conditions and interface conditions either as supplemental equations or as in Section 3.3.4. Form the system (45).
(c) Compute the QR-factorization of the matrix on the left-hand side of (45): $\overline{Q}_N = Q\,R$.
(d) Compute $c^{(*,*,*)} = R^{-1} Q^* \overline{F}_N$ where Q^* is the conjugate transpose of Q.

3. For $i = 1, 2, \ldots, N$, consider the subdomain Ω_i:

(a) Use $c^{(i,*,*)}$ obtained in step 2(d) and the series representation (25) for each $\Gamma^{(i,j)}$ to reconstruct the boundary data $\xi_{\Gamma^{(i)}}$.
(b) Extend $\xi_{\Gamma^{(i)}}$ to the grid boundary $\gamma^{(i)}$ using $\mathbf{Ex}$ given by (22): $\xi_{\gamma^{(i)}} = \mathbf{Ex}\xi_{\Gamma^{(i)}}$.
(c) Apply (16) to $\xi_{\gamma^{(i)}}$ derived in step 3(b) to obtain a local, fourth-order approximation to the solution u on the subdomain 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 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), which involves multiple solutions of the AP (11). The results of step 1(b) form the matrix $Q^{(i,*,*)}$ in step 1(c). However, $Q^{(i,*,*)}$ does not need to be recomputed every time the algorithm is run. As long as the subdomains have the same shape and the same basis functions are used 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. If k is uniform across Ω, then we have $Q^{(i,*,*)} = Q^{(j,*,*)}$ for all $i, j \in \{1, \ldots, N\}$, and the linear system (30) is only built 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(c), a QR factorization is used to find the least squares solution of the matrix equation (45). The cost of QR factorization grows as the number of subdomains increases (see Section 3.6). However, once the factorization has been performed, changes to the right-hand sides of (8a) and (8b) (i.e., f and ϕ, respectively) do not affect the left-hand side of (45). 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 4. Further, in the case where ϕ changes while f remains the same, step 1(d) can also be reused, thus starting the algorithm from step 2(d) and saving the cost of applying $\mathcal{G}^{(h)}$ in step 1(d).

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 (8b), 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.

Let us finally note that, the condition number of the matrix $\overline{Q}_N$ on the left-hand side of (45) may, in principle, depend on the total number of subdomains N, as well as the structure of decomposition, e.g., strip-type vs. checkerboard-type, see Fig. 10. Yet if there is a dependence, in our implementation it did not affect either the accuracy of the computed solution or performance of the algorithm. In particular, in Section 4.1 we report that the cost of QR factorization of the matrix $\overline{Q}_N$ (step 2(c)) scales even slower than theoretically predicted in Section 3.6. This obviates the need for any preconditioners.

3.6. Complexity

The complexity of the algorithm depends on two main factors: Solving the discrete AP (11), i.e., applying the operator $\mathcal{G}^{(h)}$, and computing the QR-factorization of $\overline{Q}_N$ from (45). Further, the applications of $\mathcal{G}^{(h)}$ include the pre-computed construction of $Q^{(i,*,*)}$ and the run-time steps 1(d) 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 paper. 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 $\mathcal{G}^{(h)}$. 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 3.2.1 the choice of boundary

conditions for the y-boundaries in (10) 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 $\mathcal{G}^{(h)}$, but $\mathcal{G}^{(h)}$ gets applied many times over the course of the method. In particular, the construction of any $Q^{(i,*,*)}$ requires $8M^*$ applications of $\mathcal{G}^{(h)}$ — 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, $\mathcal{G}^{(h)}$ is also applied twice to every subdomain to construct the right-hand side of (43), and an additional application is required in (16) to obtain the final approximation. These $3N$ applications contribute $\mathcal{O}(Nn^2 \log n)$ to the overall complexity.

The cost of the QR-factorization of $\overline{Q}_N$ from (45) is the other main contribution to the method's complexity. Assuming that (45) is formed by resolving the boundary and transmission conditions as discussed in Section 3.3.4, 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 Fig. 7c. 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 kept constant for a given collection of problems (the selection of M^* is discussed in Section 4), so we consider the cost to be $\mathcal{O}\left(N^3 n\right)$. When the number of subdomains is large, it is beneficial to avoid repeating the QR-factorization when possible as discussed in Section 3.5, 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 $\mathcal{O}\left(N^3 n + Nn^2 \log n\right)$. 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 (8) 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 those 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.

4. 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 3.6. For all the test cases, we consider the Helmholtz equation (8) where the wavenumber k is constant on any given Ω_i but piecewise constant over Ω. The coefficients α and β from (8b) 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 strip-type 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 3.6, where Ω_1 is again centered at the origin, and acts as the "bottom-left" corner of Ω (see Fig. 15), 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 (checkerboard-type decomposition).

All auxiliary problems are solved using the fourth-order accurate compact finite difference scheme (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 [23], 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 [6], 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.

Table 1
Grid convergence and time (in seconds) to apply $\mathcal{G}^{(h)}$ 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 (8b), and $M^* = 40$.

n	Error	Rate	$\mathcal{G}^{(h)}$ 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

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 (8a) and (8b), respectively. In this case, the error is computed by taking the maximum norm of the difference between the approximate and exact solution on the grid $\mathbb{N}^+$ (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 assess the 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 $\mathbb{N}^+$ 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 [14] to handle all Chebyshev polynomial related operations. The QR-factorizations were performed using MATLAB's built-in 'economy-size' QR-factorization.

4.1. Uniform wavenumber

In order to evaluate the complexity of the solver, we start with the case of one subdomain Ω. 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 (8b). The results in Table 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 3.6, 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 1 as n is doubled. Note that Table 1 also corroborates the same complexity for the construction of Q from (30) (or equivalently, $Q^{(i,*,*)}$ from (43)) because the dominating cost of constructing Q is the application of $\mathcal{G}^{(h)}$ for every basis function in the subdomain. Note that, the $\mathcal{G}^{(h)}$ 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.

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 2–5 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 (45) and its QR-factorization remain the same for all three test problems. Hence, for each of Tables 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 shows the grid convergence of the case with two subdomains, which uses the example domain given in Fig. 2 with basic Robin boundary conditions defined by $\alpha = 1$ and $\beta = 1$ in (8b). By comparing the grid convergence of the first test case in Table 2 to that of Table 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
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 (8b). Errors marked with * are computed with $M^* = 20$.

n	$e^{i\frac{k}{\sqrt{2}}(x+y)}$		$e^{\frac{-1}{1-(x^2+y^2)}}$		$\sin^4(\pi x)\sin(\pi y)$	
	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

Table 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 (8b). 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.

n	QR Time	Ratio	$e^{i\frac{k}{\sqrt{2}}(x+y)}$		$e^{\frac{-1}{1-(x^2+y^2)}}$		$\sin^4(\pi x)\sin(\pi y)$	
			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 4
Grid convergence for three test solutions where Ω is a large square comprising of 3 subdomains in each direction (checkerboard-type decomposition), 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 (8b). Errors marked with * are computed with $M^* = 20$.

n	$e^{i\frac{k}{\sqrt{2}}(x+y)}$		$e^{\frac{-1}{1-(x^2+y^2)}}$		$\sin^4(\pi x)\sin(\pi y)$	
	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

In Table 3, we use the exact same test solutions, wavenumber, and boundary conditions as in Table 2, but on a larger scale with $N = 24$ subdomains extending in the positive x-direction from Ω_1 (strip-type decomposition). 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 3.6 (see Table 6 for the complexity with respect to N). In the first two columns of Table 3, we see that as the grid dimension n is doubled, the time for the QR-factorization is approximately doubled as well.

The case of a large square domain is reported in Table 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 4, the errors and convergence rate are unaffected by the presence of an interior subdomain and cross-points in this simple case, while more difficult cases are presented in Tables 10 and 11 in Section 4.2.

Table 5 displays the grid convergence for a problem with mixed boundary conditions (see also Fig. 11), 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 5 are the same as in Table 2, saving the cost of two applications of $\mathcal{G}^{(h)}$ (per subdomain) because F in the right-hand side of (43) does not need to be computed. As the type of boundary condition changed (i.e., α or β in (8b) changed) we need to recompute the QR-factorization for the first test solution, reusing it for the second and third test solutions.

Table 5
Grid convergence for three test solutions over two subdomains with mixed boundary conditions as indicated in Fig. 11. A uniform wavenumber $k = 13$ is used in both subdomains, and $M^* = 40$. Errors marked with * are computed with $M^* = 20$.

n	$e^{i\frac{k}{\sqrt{2}}(x+y)}$		$e^{\frac{-1}{1-(x^2+y^2)}}$		$\sin^4(\pi x)\sin(\pi y)$	
	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

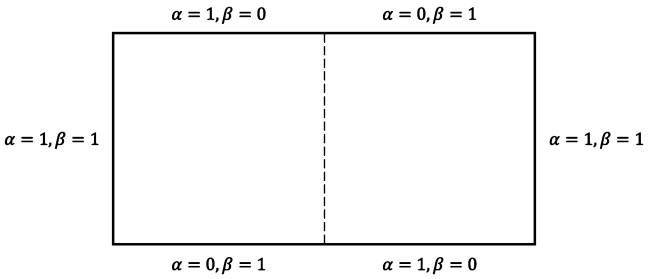


Fig. 11. The mixed boundary condition for Table 5. The coefficients α and β are defined separately for each exterior edge of the subdomains.

Table 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 (8b).

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

In Table 6, we can see how the timing for the QR-factorization grows with respect to the number of subdomains, N. Recall from Section 3.6 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 6, the execution time of the QR-factorization scales slower than its theoretical complexity would suggest, at least for the sizes of problems that we have tested.
The data in Table 6 indicate, in particular, that any potential dependence of the conditioning of system (45) on the total number of subdomains N does not hamper the performance of QR factorization and therefore, there is no need for using any preconditioners.

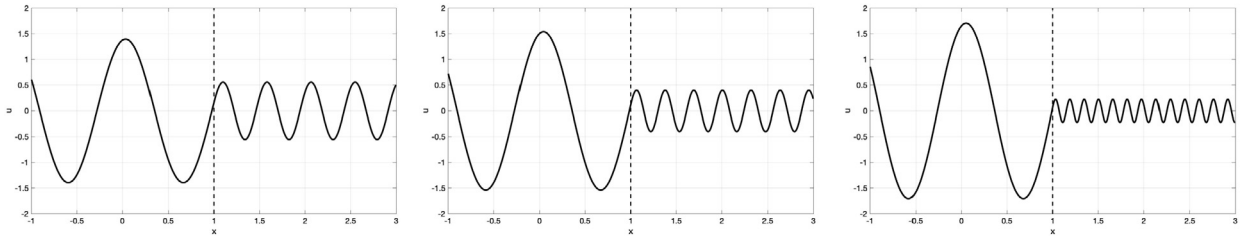
4.2. Piecewise-constant wavenumber — material discontinuities

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 7 and 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, $e^{ik_1 x}$ 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 Appendix A. Table 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 Fig. 12. The method maintains its fourth-order rate of convergence, even for the largest jump from 5 to 40, which creates a strong material discontinuity along the interface. 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

Table 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 Fig. 12. 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 $M^* = 30$.

n	$k_1 = 5$ $k_2 = 13$ $M^* = 40$		$k_1 = 5$ $k_2 = 20$ $M^* = 50$		$k_1 = 5$ $k_2 = 40$ $M^* = 60$	
	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) $k_1 = 5$, $k_2 = 40$

Fig. 12. Real part of the test solutions from Table 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 8
Grid convergence for $u = e^{ikx}$ over four subdomains (strip-type decomposition) 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

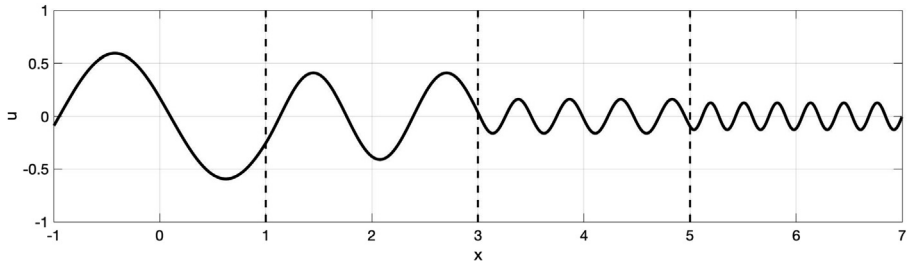


Fig. 13. Real part of the test solution from Table 8. The solution has no dependence on y, so a cross-section is plotted. Each Ω_i has a distinct wavenumber k_i, with $k_1 = 3$, $k_2 = 5$, $k_3 = 13$, and $k_4 = 20$. Moving in the positive x-direction, the wavenumber and frequency increase, while the amplitude decreases.

grid, so M^* is reduced only for the $n = 64$ grid in the relevant test cases. The four-subdomain case presented in Table 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 Fig. 13. Even with four unique wavenumbers, it can be seen in Table 8 that the method still has fourth-order convergence.

Additionally, we point out the increase in error as k_2 increases in Table 7. We attribute this increase to the pollution effect [3], because it appears consistent with our additional observations of the pollution effect for problems with uniform

Table 9
Grid convergence on the duct of $N = 16$ subdomains depicted in Fig. 14, various homogeneous boundary conditions, and the source function from (46). The wavenumbers alternate between $k = 5$ and $k = 40$, and $M^* = 60$.

n	Dirichlet		Neumann		Robin	
	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	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

Fig. 14. The decomposition used to compute Table 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.

Table 10
Grid convergence on a large square decomposed into $N = N_d^2$ subdomains and a checkerboard pattern for its wavenumber as depicted in Fig. 15. There are homogeneous Dirichlet boundary conditions and the source function is (46). The checkerboard wavenumbers are $k = 5$ (gray) and $k = 40$ (white), with $M^* = 60$.

n	$N = 16$		$N = 25$		$N = 36$	
	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	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

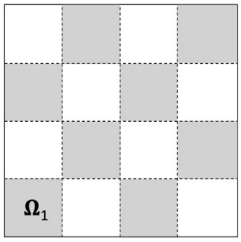


Fig. 15. The 4×4 decomposition used in the first case of Table 10 where the wavenumbers are assigned in a checkerboard pattern, with $k = 5$ in the gray subdomains and $k = 40$ in the white subdomains.

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:

$$f(x, y) = \begin{cases} \exp\left(\frac{-1}{\frac{1}{4} - (x^2 + y^2)}\right) & x^2 + y^2 < \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{46}$$

In Table 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 Fig. 14). The three cases in Table 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 10, the domain is a large square decomposed into $N = N_d^2$ (cf. Section 3.6) subdomains, where the piecewise constant values of the wavenumber are defined in a checkerboard pattern with $k = 5$ and $k = 40$ as in Fig. 15, so that there is a strong material discontinuity along every interface. These examples combine the qualitative aspects of Tables 4 and 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 either material discontinuities or cross-points (interior or boundary), yet the method's convergence does not suffer from their presence.

Table 11
Grid convergence on a square domain decomposed into $N = 9$ subdomains with homogeneous Dirichlet boundary conditions and the source function from (46). Each subdomain has a unique wavenumber as depicted in Fig. 16, 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

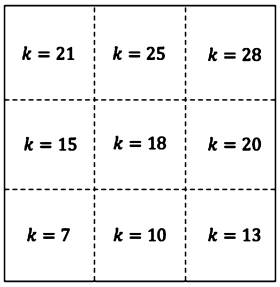


Fig. 16. Piecewise constant values of the wavenumber k for the example computed in Table 11.

Table 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 Fig. 16. In this case, there is also a material discontinuity along every interface. In contrast to the configurations of Figs. 14 and 15, this example uses a different wavenumber in each subdomain (similar to Fig. 13). This is the costliest test case for the method, as $Q^{(i,*,*)}$ is different for every Ω_i. Yet we can still use those $Q^{(i,*,*)}$ matrices computed for $k = 5, 13$, and 20 that were used for earlier examples. In Table 11, fourth-order convergence is clearly observed.

4.3. 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 so that the method converges with its design rate. Moving beyond simple square and duct decompositions will often introduce reentrant corners, an example of which is shown in the third plot of Fig. 10. 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" domain in Fig. 17 which contains one reentrant corner, and demonstrate the convergence obtained by our method under two different types of solutions.

Table 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 (46). In the first case, the test solution contains no singularities, and as expected Table 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 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 in the beginning of this section.

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 [21]. The method can also handle general shaped subdomains but these fall outside the scope of this paper.

5. Conclusions

In this paper, we have adapted the method of difference potentials to solve a non-overlapping domain decomposition formulation for the Helmholtz equation. The key distinction between the proposed approach and most traditional DDMs is that we enforce the interface conditions exactly (more precisely, with spectral accuracy) and with no use of iterations. After solving for the spectral boundary data along all boundaries and interfaces, the direct solves for all subproblems can be distributed and performed concurrently. Unlike the conventional iterative non-overlapping DDMs, the proposed formulation proves insensitive to strong material discontinuities, mixed/discontinuous boundary conditions, and cross-points in the domain. It also performs uniformly well regardless of the structure of decomposition, e.g., strip-type vs. checkerboard-type, including cases with strong material discontinuities along every interface. Numerical results corroborate the fourth-order

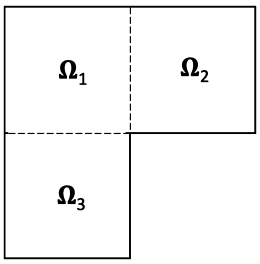


Fig. 17. The "Block L" domain decomposition used in Table 12. Ω_1 is centered over the origin, with interfaces to Ω_2 and Ω_3 at $x = 1$ and $y = -1$, respectively. This configuration creates a reentrant corner at the point $(x, y) = (1, -1)$.

Table 12
Grid convergence for the Block L case from Fig. 17 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 (46). Case 2 develops a singularity at the reentrant corner and breaks down the convergence.

n	Case 1		Case 2	
	Error	Rate	$\|u_n^{(h)} - u_{\frac{n}{2}}^{(h)}\|_\infty$	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

convergence rate of the method in numerous situations, most notably for decompositions with cross-points and large jumps in the wavenumber between neighboring subdomains. 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.

The overall cost of the method is composed of the individual subdomain solves and a global solve for the boundary/interface data. A clear advantage of the MDP is that, regardless of the actual physical boundary conditions, it allows for the use of optimal log-linear solvers for individual subdomains. Therefore, the solutions on subdomains are computed very efficiently. Moreover, those direct subdomain solves can be parallelized on a number of processors up to the number of subdomains. The representation of the boundary/interface data employed by the proposed method is spectral rather than grid-based. Its dimension is low compared to the typical grid dimensions, and the cost of computing the boundary/interface data by QR factorization starts to dominate over the individual subdomain solves for the cases with more than $N = 8$ subdomains. We emphasize that, it is precisely this global QR solve for the coefficients of the spectral expansion at the boundary that guarantees the uniform performance of the method for the cases with cross-points and strong material discontinuities. Moreover, once the QR factorization has been computed, new problems with variations in the source and boundary data can be solved at a substantially reduced cost by simply reusing the previously computed QR factorization. In other words, one QR factorization allows one to solve an entire class of similar yet not identical problems.

The framework that is laid out in this paper can be adapted and extended in numerous ways to broaden the applicability of the method. Briefly mentioned in Section 3.3.3, transmission conditions of the form (40) 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.

Appendix A. Derivation of a function with piecewise constant wavenumber

Consider a domain $\Omega \subset \mathbb{R}^2$ split into two subdomains, Ω_1 and Ω_2, as in Fig. A.18. 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.

Let $u_1 = e^{ik_1 x}$ 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

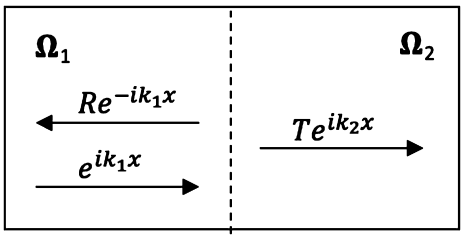


Fig. A.18. The setup for deriving the reflected and transmitted parts of a one-dimensional incident wave hitting an interface in a domain composed of two subdomains.

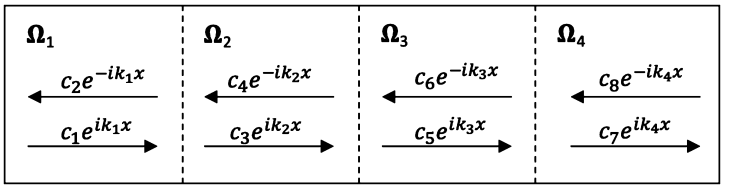


Fig. A.19. The setup for deriving the reflected and transmitted parts of a one-dimensional incident wave hitting an interface in a domain composed of four subdomains.

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:

$$u(x, y) = \begin{cases} e^{ik_1x} + Re^{-ik_1x}, & x \leqslant 0 \\ Te^{ik_2x}, & x \geqslant 0 \end{cases} \tag{A.1}$$

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

$$\begin{aligned} &e^{ik_1x}|_{x=0} + Re^{-ik_1x}|_{x=0} = Te^{ik_2x}|_{x=0} \\ &\qquad \Longrightarrow 1 + R = T \end{aligned} \tag{A.2}$$

and for continuity of the derivative we get

$$\begin{aligned} &\left(\frac{\partial}{\partial x} e^{ik_1x}\right)|_{x=0} + \left(\frac{\partial}{\partial x} Re^{-ik_1x}\right)|_{x=0} = \left(\frac{\partial}{\partial x} Te^{ik_2x}\right)|_{x=0} \\ \Longrightarrow &\left(ik_1 e^{ik_1x}\right)|_{x=0} - \left(ik_1 Re^{-ik_1x}\right)|_{x=0} = \left(ik_2 Te^{ik_2x}\right)|_{x=0} \\ &\qquad \Longrightarrow ik_1 - ik_1 R = ik_2 T \\ &\qquad \Longrightarrow k_1(1 - R) = k_2 T \\ &\qquad \Longrightarrow 1 - R = \frac{k_2}{k_1} T \end{aligned} \tag{A.3}$$

By combining (A.2) and (A.3), we can solve for R and T to get

$$R = \frac{k_2}{2k_1} - \frac{1}{2} \qquad T = \frac{k_2}{2k_1} + \frac{1}{2} \tag{A.4}$$

The values of R and T from (A.4) can be plugged into (A.1) to obtain the function $u(x, y)$, defined across Ω.

For a larger case with four subdomains, consider the scenario depicted in Fig. A.19, 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:

$$c_1 e^{ik_1x_1} + c_2 e^{-ik_1x_1} = c_3 e^{ik_2x_1} + c_4 e^{-ik_2x_1}$$

which can be rewritten as

$$c_1 e^{ik_1x_1} + c_2 e^{-ik_1x_1} - c_3 e^{ik_2x_1} - c_4 e^{-ik_2x_1} = 0$$

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

$$
\begin{aligned}
c_1 e^{ik_1 x_1} + c_2 e^{-ik_1 x_1} - c_3 e^{ik_2 x_1} - c_4 e^{-ik_2 x_1} &= 0 \\
c_3 e^{ik_2 x_2} + c_4 e^{-ik_2 x_2} - c_5 e^{ik_3 x_2} - c_6 e^{-ik_3 x_2} &= 0 \\
c_5 e^{ik_3 x_3} + c_6 e^{-ik_3 x_3} - c_7 e^{ik_4 x_3} - c_8 e^{-ik_4 x_3} &= 0 \\
k_1 c_1 e^{ik_1 x_1} - k_1 c_2 e^{-ik_1 x_1} - k_2 c_3 e^{ik_2 x_1} + k_2 c_4 e^{-ik_2 x_1} &= 0 \\
k_2 c_3 e^{ik_2 x_2} - k_2 c_4 e^{-ik_2 x_2} - k_3 c_5 e^{ik_3 x_2} + k_3 c_6 e^{-ik_3 x_2} &= 0 \\
k_3 c_5 e^{ik_3 x_3} - k_3 c_6 e^{-ik_3 x_3} - k_4 c_7 e^{ik_4 x_3} + k_4 c_8 e^{-ik_4 x_3} &= 0
\end{aligned}
$$

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 $c_7 e^{ik_4 x}$ 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.

References

[1] I.M. Babuska, S.A. Sauter, Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?, SIAM Rev. 42 (2000) 451–484.
[2] A. Bayliss, C. Goldstein, E. Turkel, An iterative method for the Helmholtz equation, J. Comput. Phys. 49 (1983) 443–457, https://doi.org/10.1016/0021-9991(83)90139-0.
[3] A. Bayliss, C.I. Goldstein, E. Turkel, On accuracy conditions for the numerical computation of waves, J. Comput. Phys. 59 (1985) 396–404.
[4] Y. Boubendir, D. Midura, Non-overlapping domain decomposition algorithm based on modified transmission conditions for the Helmholtz equation, Comput. Math. Appl. 75 (2018) 1900–1911.
[5] Y. Boubendir, X. Antoine, C. Geuzaine, A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation, J. Comput. Phys. 231 (2012) 262–280.
[6] D.S. Britt, S.V. Tsynkov, E. Turkel, A high-order numerical method for the Helmholtz equation with nonstandard boundary conditions, SIAM J. Sci. Comput. 35 (2013) A2255–A2292, https://doi.org/10.1137/120902689.
[7] S. Britt, S. Tsynkov, E. Turkel, Numerical simulation of time-harmonic waves in inhomogeneous media using compact high order schemes, Commun. Comput. Phys. 9 (2011) 520–541, https://doi.org/10.4208/cicp.091209.080410s.
[8] A.P. Calderon, Boundary-value problems for elliptic equations, in: Proceedings of the Soviet-American Conference on Partial Differential Equations in Novosibirsk, Fizmatgiz, Moscow, 1963, pp. 303–304.
[9] X. Claeys, E. Parolin, Robust treatment of cross points in optimized Schwarz methods, arXiv:2003.06657, 2020.
[10] Xavier Claeys, Non-local variant of the optimized Schwarz method for arbitrary non-overlapping subdomain partitions, ESAIM: M2AN 55 (2021) 429–448, https://doi.org/10.1051/m2an/2020083.
[11] A. Deraemaeker, I. Babuska, P. Bouillard, Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions, Int. J. Numer. Methods Eng. 46 (1999) 471–499.
[12] B. Després, Domain decomposition method and the Helmholtz problem. II, in: Second International Conference on Mathematical and Numerical Aspects of Wave Propag., 1993, pp. 197–206.
[13] V. Dolean, P. Jolivet, F. Nataf, An introduction to domain decomposition methods, in: Algorithms, Theory, and Parallel Implementation, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015, https://doi-org.prox.lib.ncsu.edu/10.1137/1.9781611974065.ch1.
[14] T. Driscoll, N. Hale, L. Trefethen, Chebfun Guide, Pafnuty Publications, Oxford, 2014.
[15] Y. Epshteyn, Q. Xia, Efficient numerical algorithms based on difference potentials for chemotaxis systems in 3D, J. Sci. Comput. 80 (2019) 26–59, https://doi.org/10.1007/s10915-019-00928-z.
[16] M. Gander, K. Santugini, Cross-points in domain decomposition methods with a finite element discretization, Electron. Trans. Numer. Anal. 45 (2016) 219–240.
[17] M.J. Gander, Schwarz methods over the course of time, Electron. Trans. Numer. Anal. 31 (2008) 228–255.
[18] D. Gordon, R. Gordon, CADD: a seamless solution to the domain decomposition problem of subdomain boundaries and cross-points, Wave Motion 98 (2020) 102649, https://doi.org/10.1016/j.wavemoti.2020.102649.
[19] I. Harari, E. Turkel, Accurate finite difference methods for time-harmonic wave propagation, J. Comput. Phys. 119 (1995) 252–270, https://doi.org/10.1006/jcph.1995.1134.
[20] P.-L. Lions, On the Schwarz alternating method III: a variant for nonoverlapping subdomains, 1990.
[21] S. Magura, S. Petropavlovsky, S. Tsynkov, E. Turkel, High-order numerical solution of the Helmholtz equation for domains with reentrant corners, Appl. Numer. Math. 118 (2017) 87–116.
[22] V. Mattesi, M. Darbas, C. Geuzaine, A quasi-optimal non-overlapping domain decomposition method for two-dimensional time-harmonic elastic wave problems, J. Comput. Phys. (2019) 109050.
[23] M. Medvinsky, S. Tsynkov, E. Turkel, The method of difference potentials for the Helmholtz equation using compact high order schemes, J. Sci. Comput. 53 (2012) 150–193.
[24] M. Medvinsky, S. Tsynkov, E. Turkel, Solving the Helmholtz equation for general smooth geometry using simple grids, Wave Motion 62 (2016) 75–97, https://doi.org/10.1016/j.wavemoti.2015.12.004.
[25] A. Modave, A. Royer, X. Antoine, C. Geuzaine, An optimized Schwarz domain decomposition method with cross-point treatment for time-harmonic acoustic scattering, HAL-02432422, 2020.
[26] J. Moshfegh, A parallel direct method for finite element electromagnetic computations based on domain decomposition, Ph.D. thesis, University of Massachusetts Amherst, 2019, https://scholarworks.umass.edu/dissertations_2/1800.

[27] J. Moshfegh, M.N. Vouvakis, Direct domain decomposition method (D^3M) for finite element electromagnetic computations, in: 2016 IEEE International Symposium on Antennas and Propagation, APSURSI, 2016, pp. 1127–1128.
[28] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford Science Publications, 1999.
[29] A.A. Reznik, Approximation of surface potentials of elliptic operators by difference potentials, Sov. Math. Dokl. 25 (1982) 543–545.
[30] V.S. Ryaben'kii, Boundary equations with projections, Russ. Math. Surv. 40 (1985) 147–183.
[31] V.S. Ryaben'kii, Method of Difference Potentials and Its Applications, Springer Series in Computational Mathematics, vol. 30, Springer-Verlag, Berlin, 2002.
[32] V.S. Ryaben'kii, S.V. Tsynkov, A Theoretical Introduction to Numerical Analysis, Chapman & Hall/CRC, Boca Raton, FL, 2007.
[33] H.A. Schwarz, Über einen grenzübergang durch alternierendes verfahren, Vierteljahrsschr. Nat.forsch. Ges. Zür. 15 (1870) 272–286.
[34] R.T. Seeley, Singular integrals and boundary value problems, Am. J. Math. 88 (1966) 781–809.
[35] I. Singer, E. Turkel, High-order finite difference methods for the Helmholtz equation, Comput. Methods Appl. Mech. Eng. 163 (1998) 343–358, https://doi.org/10.1016/s0045-7825(98)00023-1.
[36] I. Singer, E. Turkel, Sixth-order accurate finite difference schemes for the Helmholtz equation, J. Comput. Acoust. 14 (2011), https://doi.org/10.1142/S0218396X06003050.
[37] C.C. Stolk, A rapidly converging domain decomposition method for the Helmholtz equation, J. Comput. Phys. 241 (2013) 240–252.
[38] A. Toselli, O. Widlund, Domain Decomposition Methods—Algorithms and Theory, Springer Series in Computational Mathematics, vol. 34, Springer-Verlag, Berlin, 2005.
[39] E. Turkel, D. Gordon, R. Gordon, S. Tsynkov, Compact 2d and 3d sixth order schemes for the Helmholtz equation with variable wave number, J. Comput. Phys. 232 (2013) 272–287.