

A Class of Iterative Solvers for the Helmholtz Equation: Factorizations, Sweeping Preconditioners, Source Transfer, Single Layer Potentials, Polarized Traces, and Optimized Schwarz Methods*

Martin J. Gander[†]
Hui Zhang[‡]

Abstract. Solving time-harmonic wave propagation problems by iterative methods is a difficult task, and over the last two decades an important research effort has gone into developing preconditioners for the simplest representative of such wave propagation problems, the Helmholtz equation. A specific class of these new preconditioners is considered here. They were developed by researchers with various backgrounds using formulations and notations that are very different, and all are among the most promising preconditioners for the Helmholtz equation.

The goal of the present article is to show that this class of preconditioners is based on a common mathematical principle, and that they can all be formulated in the context of domain decomposition methods known as optimized Schwarz methods. This common formulation allows us to explain in detail how and why all these methods work. The domain decomposition formulation also allows us to avoid technicalities in the implementation description we give of these recent methods.

The equivalence of these methods with optimized Schwarz methods translates at the discrete level into equivalence with approximate block LU decomposition preconditioners, and in each case we give the algebraic version, including a detailed description of the approximations used. While we choose to use the Helmholtz equation for which these methods were developed, our notation is completely general and the algorithms we give are written for an arbitrary second-order elliptic operator. The algebraic versions are even more general, assuming only a connectivity pattern in the discretization matrix.

All the new methods studied here are based on sequential decomposition of the problem in space into a sequence of subproblems, and they have in their optimal form the property to lead to nilpotent iterations, like an exact block LU factorization. Using our domain decomposition formulation, we finally present an algorithm for two-dimensional decompositions, i.e., decompositions that contain cross points, which is still nilpotent in its optimal form. Its approximation is currently an active area of research, and it would have been difficult to discover such an algorithm without the domain decomposition framework.

Key words. Helmholtz, iterative, preconditioner, factorization, sweeping, Schwarz

AMS subject classifications. 65N55, 65N22, 65F10

*Received by the editors October 7, 2016; accepted for publication (in revised form) January 29, 2018; published electronically February 7, 2019.

<http://www.siam.org/journals/sirev/61-1/M109781.html>

Funding: The work of the first author was supported by the Swiss National Science Foundation SNF (200020_168999/1). The work of the second author was supported by Zhejiang Provincial Natural Science Foundation of China (LY17A010014), Natural Science Foundation of China (11671074, 11771313, 11771398), and research start funding of Zhejiang Ocean University.

[†]Section of Mathematics, University of Geneva, 1211 Geneva 4, Switzerland (martin.gander@unige.ch).

[‡]Corresponding author. School of Mathematics and Statistics, Northeast Normal University, Changchun 130024, China (mike.hui.zhang@hotmail.com).

Contents	
I Introduction	4
2 Underlying Fundamental Algorithms	5
3 The Helmholtz Equation	8
4 Direct and Iterative Solvers	11
5 Notation	14
5.1 Domain Decomposition	15
5.2 Transmission Conditions	17
5.3 Green's Function	22
6 Optimized Schwarz Methods	24
6.1 Subdomain Transmission Form of Schwarz Methods	24
6.2 Global Deferred Correction Form of Schwarz Methods	28
6.3 Substructured Form of Schwarz Methods	33
7 AILU and Sweeping Preconditioners	35
8 Methods Motivated by Physics	41
8.1 The Source Transfer Method Using Equivalent Volume Sources	43
8.2 The Method Using Single Layer Potentials	49
8.3 Method of Polarized Traces Using Single and Double Layer Potentials	55
9 Optimal Parallel Schwarz Methods for Arbitrary Decompositions	61
10 Numerical Experiments	63
11 Conclusions	66
References	68

I. Introduction. Solving the Helmholtz equation numerically for moderate to high wavenumbers is a difficult task, and very different from solving Laplace-like problems. This is due to three main reasons: First, Helmholtz problems are often posed on unbounded domains, which have to be artificially truncated to perform computations on finite computers, or by using Green's functions; see, e.g., [21, 134]. Second, one needs to have a much higher mesh resolution than what would usually be required to represent the solution because of the so-called pollution effect [5, 106, 182]. Finally, one must then solve the resulting very large scale system of linear equations, for which classical iterative methods are not suitable [55, 59]. Our focus here is on a class of recently developed novel iterative solvers for the Helmholtz equation based on sequential decompositions in space. Many research groups around the world have focused on developing such solvers, and for practitioners, and even specialists, it is

not easy to keep up with these developments and to grasp important novelties in these methods. The main reason for this is that these new methods are sometimes formulated at the continuous level, sometimes at the discrete level, sometimes using integral formulations, and sometimes using volume discretizations, and the groups developing these methods come from different backgrounds and use different motivations, intuitions, and notation when formulating their methods. Furthermore, most of these new methods require absorbing boundary conditions or perfectly matched layers for their formulation, which are ingredients that are not commonly encountered in classical iterative methods for Laplace-like problems.

The purpose of this article is to first describe in simple terms the main underlying fundamental algorithms for the new class of methods based on sequential decompositions in space. At the continuous level, the underlying algorithms are optimal and optimized Schwarz methods, and at the discrete level, the underlying algorithms are exact and approximate block LU factorizations. This first, relatively short part gives the main insight needed to understand the new Helmholtz solvers in a simplified and nontechnical setting. In the second, main part, we then rigorously show how members of this new class of Helmholtz solvers are tightly related. To do so, we introduce a compact notation that allows us to formulate all these new techniques, and we give for each one first the original formulation given by the authors, and then equivalent formulations at the continuous and discrete levels in the form of the fundamental underlying algorithms, for which we can prove equivalence results. We hope that our article will help people working in this challenging area of numerical analysis to rapidly understand this new class of algorithms and their potential.

2. Underlying Fundamental Algorithms. We start by explaining two fundamental algorithms which are very much related, one at the discrete level and one at the continuous level. These algorithms are the key ingredient in all the recent iterative methods proposed for the Helmholtz equation. Even though these algorithms can be formulated for other partial differential equations (PDEs) and all our equivalence results still hold, we use first the Helmholtz equation in its simplest form to explain them, namely,

$$(1) \quad (\Delta + k^2)u = f \quad \text{in } \Omega := (0, 1) \times (0, 1),$$

with suitable boundary conditions to make the problem well-posed.¹ Discretizing (1) using a standard five point finite difference discretization for the Laplacian on an equidistant grid leads to the linear system of equations

$$(2) \quad \begin{bmatrix} D_1 & U_1 & & & \\ L_1 & D_2 & U_2 & & \\ & \ddots & \ddots & \ddots & \\ & & L_{J-2} & D_{J-1} & U_{J-1} \\ & & & L_{J-1} & D_J \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{J-1} \\ u_J \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{J-1} \\ f_J \end{bmatrix},$$

¹We use this simplest form of the Helmholtz equation to start with, then treat in the main part of the article the more complete formulation given in (11).

where $D_j = \text{tridiag}(\frac{1}{h^2}, -\frac{4}{h^2} + k^2, \frac{1}{h^2})$,² $L_j = U_j = \text{diag}(\frac{1}{h^2})$. The block LU factorization of the coefficient matrix in (2) is given by

$$(3) \quad A = \begin{bmatrix} T_1 & & & & \\ L_1 & T_2 & & & \\ & \ddots & \ddots & & \\ & & L_{J-2} & T_{J-1} & \\ & & & L_{J-1} & T_J \end{bmatrix} \begin{bmatrix} I_1 & T_1^{-1}U_1 & & & \\ & I_2 & T_2^{-1}U_2 & & \\ & & \ddots & \ddots & \\ & & & I_{J-1} & T_{J-1}^{-1}U_{J-1} \\ & & & & I_J \end{bmatrix},$$

where the T_j 's are the Schur complements³ that satisfy the recurrence relation

$$(4) \quad T_1 = D_1, \quad T_j = D_j - L_{j-1}T_{j-1}^{-1}U_{j-1} \quad \text{for } j \geq 2,$$

as one can see by simply multiplying the two factors in (3) and comparing with the original matrix in (2). Using this factorization, we can solve (2) by first solving by forward substitution the block lower triangular system

$$(5) \quad \begin{bmatrix} T_1 & & & & \\ L_1 & T_2 & & & \\ & \ddots & \ddots & & \\ & & L_{J-2} & T_{J-1} & \\ & & & L_{J-1} & T_J \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{J-1} \\ \mathbf{v}_J \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_{J-1} \\ \mathbf{f}_J \end{bmatrix},$$

and then solving by backward substitution the block upper triangular system

$$(6) \quad \begin{bmatrix} I_1 & T_1^{-1}U_1 & & & \\ & I_2 & T_2^{-1}U_2 & & \\ & & \ddots & \ddots & \\ & & & I_{J-1} & T_{J-1}^{-1}U_{J-1} \\ & & & & I_J \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{J-1} \\ \mathbf{u}_J \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{J-1} \\ \mathbf{v}_J \end{bmatrix}.$$

This shows that one forward sweep (forward substitution) and one backward sweep (backward substitution) are enough to solve the linear system, and this is the fundamental underlying idea of the new “sweeping algorithms” for the Helmholtz equation mentioned in the title. This becomes a preconditioner if the block LU factorization is approximated by using approximate Schur complement matrices instead of the exact matrices. If we use the exact matrices, then the iteration converges in one step and thus the iteration matrix is nilpotent of degree (or index) one. One can, however, already see another one of the new algorithms here by taking a closer look at the forward substitution in (5): solving the first equation, and substituting the result into the second one, and the result obtained into the third one, and so on, we find

$$(7) \quad \begin{aligned} \mathbf{v}_1 &= T_1^{-1}\mathbf{f}_1, \\ \mathbf{v}_2 &= T_2^{-1}(\mathbf{f}_2 - L_1\mathbf{v}_1) = T_2^{-1}(\mathbf{f}_2 - L_1T_1^{-1}\mathbf{f}_1) =: T_2^{-1}\tilde{\mathbf{f}}_2, \\ \mathbf{v}_3 &= T_3^{-1}(\mathbf{f}_3 - L_2\mathbf{v}_2) = T_3^{-1}(\mathbf{f}_3 - L_2T_2^{-1}\tilde{\mathbf{f}}_2) =: T_3^{-1}\tilde{\mathbf{f}}_3, \\ &\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \end{aligned}$$

²We assume homogeneous Dirichlet boundary conditions and well-posedness for simplicity to start with; see section 4 for more information.

³For now, we also assume here for simplicity that the T_j 's are invertible.

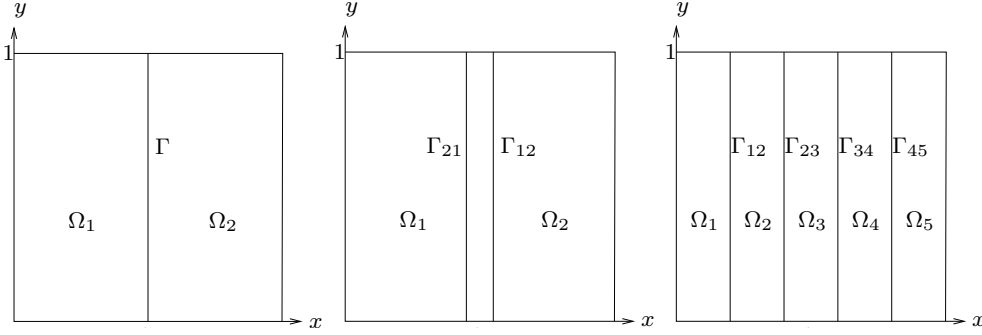


Fig. 1 Domain decompositions of the unit square: without or with overlap, and many subdomains.

where we introduced new source terms $\tilde{f}_2 := f_2 - L_1 T_1^{-1} f_1$, $\tilde{f}_3 := f_3 - L_2 T_2^{-1} \tilde{f}_2, \dots$ to make the solve for $\mathbf{v}_2, \mathbf{v}_3, \dots$ look like the first solve for \mathbf{v}_1 . These new source terms contain a transferred source term from the previous line,

$$\tilde{f}_j := f_j - L_{j-1} T_{j-1}^{-1} \tilde{f}_{j-1},$$

which is the feature that led to the so-called source transfer methods mentioned in the title. Note that $\mathbf{v}_J = \mathbf{u}_J$, so after the forward substitution, the last set of unknowns is already the exact solution, a property that will be used later by some algorithms.

In the form in which we presented the block LU decomposition, the diagonal blocks only contained one grid line of unknowns, but one could also collect several grid lines into one block. This suggests looking at the problem at the continuous level, where we decompose the domain into subdomains, as illustrated in Figure 1. For the first decomposition on the left, which would correspond to only two large diagonal blocks at the discrete level, an optimal parallel Schwarz algorithm is given by the iterative method (see [53, 133]) for arbitrary initial guess u_1^0, u_2^0 ,

$$(8) \quad \begin{aligned} (\Delta + k^2)u_1^n &= f && \text{in } \Omega_1, \\ \partial_{\mathbf{n}_1} u_1^n + \text{DtN}_1(u_1^n) &= \partial_{\mathbf{n}_1} u_2^{n-1} + \text{DtN}_1(u_2^{n-1}) && \text{on } \Gamma, \\ (\Delta + k^2)u_2^n &= f && \text{in } \Omega_2, \\ \partial_{\mathbf{n}_2} u_2^n + \text{DtN}_2(u_2^n) &= \partial_{\mathbf{n}_2} u_1^{n-1} + \text{DtN}_2(u_1^{n-1}) && \text{on } \Gamma, \end{aligned}$$

where $\partial_{\mathbf{n}_j}$ denotes the outward normal derivative for subdomain Ω_j , and DtN_j is the *Dirichlet-to-Neumann* (DtN) operator taking Dirichlet interface data on the interface Γ and returning the outward normal derivative of the corresponding solution of the Helmholtz equation on the exterior of the subdomain Ω_j . This algorithm converges in two iterations, and thus the iteration operator is nilpotent of degree two, as one can easily understand as follows: Denoting by $e_j^n := u - u_j^n$ the error at iteration n , this error satisfies by linearity the same equation as u_j^n , but with zero right hand side (r.h.s.), $f = 0$. Since, after the first iteration, the error e_j^1 satisfies the equation in subdomain Ω_j , its normal derivative at the interface will be exactly canceled by the DtN operator result when evaluating the r.h.s. on the interface Γ for the second iteration.⁴ The error e_j^2 then satisfies the equation in Ω_j with homogeneous data and thus by uniqueness is zero, and so the algorithm has converged.

⁴The r.h.s. on the interface is in fact an exact or transparent boundary condition for the neighboring subdomain.

The optimal parallel Schwarz algorithm in (8) can also be run with overlap, as indicated in Figure 1 (middle), i.e.,

$$(9) \quad \begin{aligned} (\Delta + k^2)u_1^n &= f && \text{in } \Omega_1, \\ \partial_{\mathbf{n}_1} u_1^n + \text{DtN}_1(u_1^n) &= \partial_{\mathbf{n}_1} u_2^{n-1} + \text{DtN}_1(u_2^{n-1}) && \text{on } \Gamma_{12}, \\ (\Delta + k^2)u_2^n &= f && \text{in } \Omega_2, \\ \partial_{\mathbf{n}_2} u_2^n + \text{DtN}_2(u_2^n) &= \partial_{\mathbf{n}_2} u_1^{n-1} + \text{DtN}_2(u_1^{n-1}) && \text{on } \Gamma_{21}. \end{aligned}$$

The overlap has no influence on the two-step convergence property of the optimal parallel Schwarz method.⁵ With J subdomains, as indicated in Figure 1 (right), the corresponding optimal parallel Schwarz algorithm

$$(10) \quad \begin{aligned} (\Delta + k^2)u_j^n &= f && \text{in } \Omega_j, \\ \partial_{\mathbf{n}_j} u_j^n + \text{DtN}_j(u_j^n) &= \partial_{\mathbf{n}_j} u_{j+1}^{n-1} + \text{DtN}_j(u_{j+1}^{n-1}) && \text{on } \Gamma_{j,j+1}, \\ \partial_{\mathbf{n}_j} u_j^n + \text{DtN}_j(u_j^n) &= \partial_{\mathbf{n}_j} u_{j-1}^{n-1} + \text{DtN}_j(u_{j-1}^{n-1}) && \text{on } \Gamma_{j,j-1}, \end{aligned}$$

converges in J iterations [133], and thus the iteration operator is nilpotent of degree J . At the discrete level, this result was also presented by F.-X. Roux at the IMACS conference in 2001. If, however, we organize the solves in sweeps, starting on the leftmost subdomain and going to the rightmost subdomain and back, i.e., sweeping once forward and once backward, the algorithm converges in one such double sweep, independent of the number of subdomains J , and thus the sweeping iteration operator becomes nilpotent of degree one. This algorithm is in fact the continuous analogue of the block LU factorization, with just the small modification that the DtN transmission condition on the right is replaced by the Dirichlet condition, as we will see later in section 7. Optimized Schwarz methods use approximations of the DtN operator and thus approximations of the transparent boundary condition, in the same spirit that approximate block LU factorizations use approximations to the Schur complement matrices. Well-known approximations of this type are *absorbing boundary conditions* (ABCs; cf. [49, 50, 96, 101]) and *perfectly matched layers* (PMLs; cf. [13, 31, 44, 99]).

3. The Helmholtz Equation. To be able to explain the details of recent iterative solvers for the Helmholtz equation, we need to consider a slightly more general Helmholtz equation than the simple model problem in (1), namely,

$$(11) \quad \mathcal{L}u := -\nabla^T(\alpha \nabla u) - \frac{\omega^2}{\kappa} u = f \text{ in } \Omega, \quad \mathcal{B}u = g \text{ on } \partial\Omega,$$

where $\omega \in \mathbb{C}$, $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is a bounded Lipschitz domain, the coefficient matrix α , the scalar field κ , and the source f are all given complex-valued quantities varying on Ω , and the unknown function u on Ω is to be sought. In addition to the truly physical part, the domain Ω may also contain artificial layers, for example, representing PMLs. The boundary condition $\mathcal{B}u = g$ is piecewise defined on $\partial\Omega = \bar{\Gamma}_D \cup \bar{\Gamma}_R \cup \bar{\Gamma}_V$ as follows:⁶

$$(12) \quad u = g_D \quad \text{on } \Gamma_D,$$

$$(13) \quad \mathbf{n}^T(\alpha \nabla u) + p_0 u = g_R \quad \text{on } \Gamma_R,$$

$$(14) \quad \mathbf{n}^T(\alpha \nabla u) + p_0 u + p_1 \mathbf{n}^T(\alpha \nabla_S u) - \nabla_S^T(q_1 \Pi_S(\alpha \mathbf{n})u + p_2 \alpha \nabla_S u) = g_V \quad \text{on } \Gamma_V,$$

⁵This will be different if one uses approximations of the DtN operators, as we will see.

⁶For simplicity, we assume either $\Gamma_V = \emptyset$ or $\Gamma_R = \emptyset$ and that Γ_V has no edges and corners.

where \mathbf{n} is the unit outer normal vector, ∇_S is the surface gradient, p_0, p_1, q_1, p_2 are complex-valued functions, and Π_S is the orthogonal projection onto the tangential plane of the surface. Besides acoustic waves, the Helmholtz equation (11) is also used to describe electromagnetics, where it can be derived from Maxwell's equations; see [134].

Example 1. In a typical geophysical application, Ω is a rectangle in \mathbb{R}^2 or a box in \mathbb{R}^3 , obtained from truncation of the free space problem. The original time domain equation in free space is given by

$$\frac{1}{c^2} \frac{\partial^2 \tilde{u}}{\partial t^2} - \rho \nabla_{\mathbf{x}}^T \left(\frac{1}{\rho} \nabla_{\mathbf{x}} \tilde{u} \right) = \tilde{f},$$

where c is the wave speed and ρ is the mass density, both dependent only on space, \tilde{u} is the pressure, and \tilde{f} is the space-time source term with compact support in $\Omega^{phy} \subset \Omega$ at all times. Ω^{phy} is the domain in which the solution is of interest. We make the ansatz that $\tilde{f}(\mathbf{x}, t)$ is a superposition of the time-harmonic sources $\rho(\mathbf{x})f(\mathbf{x}, \omega)e^{-i\omega t}$. Then, for each ω , the corresponding mode $u(\mathbf{x}, \omega)e^{-i\omega t}$ satisfies

$$-\frac{\omega^2}{\rho c^2} u - \nabla_{\mathbf{x}}^T \left(\frac{1}{\rho} \nabla_{\mathbf{x}} u \right) = f.$$

The time domain solution $\tilde{u}(\mathbf{x}, t)$ is the sum of the time-harmonic modes $u(x, \omega)$ over all possible values of ω . Note that ω is the time *frequency*, $k := \frac{\omega}{c}$ is called the *wavenumber*, and the *wavelength* is $\lambda = \frac{2\pi}{k}$. A certain boundary condition along infinity is imposed to exclude energy incoming from infinity and to allow energy outgoing to infinity, viewed from Ω^{phy} . An example is the Sommerfeld radiation condition (cf. [154, p. 189]) in a homogeneous medium,

$$(15) \quad \lim_{|\mathbf{x}| \rightarrow \infty} |\mathbf{x}|^{\frac{d-1}{2}} (\partial_{|\mathbf{x}|} u - i k u) = 0.$$

Since we are interested in only the near-field solution (i.e., in Ω^{phy}), the free space model is truncated to Ω^{phy} by imposing on $\partial\Omega^{phy}$ artificial boundary conditions or artificial layers, which leads to (11).

Example 2. Some models have physical boundaries from special materials, which leads directly to a boundary condition, e.g., Dirichlet for sound soft and Neumann for sound hard matter in acoustics. As a simple model, one can consider a parallel pipe open in one dimension and closed with walls in the other dimensions on which Dirichlet, Neumann, or artificial boundary conditions (layers) are imposed. We further truncate the open dimension to obtain (11). The truncated dimension is typically still much larger than the other dimensions such as for optical waveguides; see [125].

Example 3. An important class of models is that of the so-called scattering problems which are posed on an unbounded domain exterior to obstacles delimited by physical boundaries. A given incident wave then hits the obstacles and is scattered. The sum of the incident wave and the scattered wave gives the total wave field which satisfies homogeneous Dirichlet, Neumann, or impedance boundary conditions as shown in (12) and (13) on the physical boundaries. The scattered wave field satisfies the homogeneous Helmholtz equation and some condition along infinity. Usually, this is the Sommerfeld radiation condition given in (15) based on the assumption that the

medium is homogeneous outside a bounded region. The unbounded domain is truncated to a bounded region near the obstacles, which results in (11). Once the Dirichlet and Neumann traces of the solution are known on some surface, the solution in the far-field, i.e., far away from the obstacles, can be recovered using a representation formula; see [134].

Remark 4. The algorithms to be discussed in this paper are applicable not only to the model (11) but also to more complicated cases as long as the PDE and the boundary conditions are defined locally in space such that they make sense in subsets of Ω and $\partial\Omega$. For instance, we can supplement (14) with edge and corner conditions (see [7]) or use high-order absorbing boundary conditions localized with auxiliary unknowns (see [33, 97]) which can also be viewed as semidiscretized PMLs (see [45, 92]), and solve other PDEs (see, e.g., [130]). This will become clearer in the following sections.

We will occasionally need the weak formulation of (11) in appropriate function spaces; see, e.g., Lemma 10. Multiplying both sides of (11) with the complex conjugate of an arbitrary function v and integrating by parts in Ω , we find formally

$$\int_{\Omega} (\alpha \nabla u)^T \nabla \bar{v} - \frac{\omega^2}{\kappa} u \bar{v} - \int_{\partial\Omega} \mathbf{n}^T (\alpha \nabla u) \bar{v} = \int_{\Omega} f \bar{v}.$$

Substituting the boundary conditions from (12)–(14) into the above equation leads us to the following weak formulation of (11):

$$(16) \quad \text{find } u - \mathcal{E}g_D \in V, \text{ such that } a(u, v) + b(u, v) = c(v) \quad \forall v \in V,$$

where $\mathcal{E}g_D$ is an extension of g_D and

$$\begin{aligned} V &:= \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0, v|_{\Gamma_V} \in H^1(\Gamma_V)\}, \quad \|v\|_V = \sqrt{\|v\|_{H^1(\Omega)}^2 + \|v\|_{H^1(\Gamma_V)}^2}, \\ a(u, v) &:= \int_{\Omega} (\alpha \nabla u)^T \nabla \bar{v} - \frac{\omega^2}{\kappa} u \bar{v}, \\ b(u, v) &:= \int_{\Gamma_R \cup \Gamma_V} p_0 u \bar{v} + \int_{\Gamma_V} p_1 \mathbf{n}^T (\alpha \nabla_S u) \bar{v} + \int_{\Gamma_V} (q_1 \Pi_S(\alpha \mathbf{n})u + p_2 \alpha \nabla_S u)^T \nabla_S \bar{v}, \\ c(v) &:= V' \langle f, v \rangle_V + {}_{H^{-\frac{1}{2}}} \langle g_R, v|_{\Gamma_R} \rangle_{H^{\frac{1}{2}}} + {}_{H^{-\frac{1}{2}}} \langle g_V, v|_{\Gamma_V} \rangle_{H^{\frac{1}{2}}}. \end{aligned}$$

The well-posedness of (16) can be ensured by the following simultaneous assumptions:

- (i) $\alpha, \frac{\omega^2}{\kappa}$ are uniformly bounded in Ω , and so are p_0, p_1, q_1, p_2 in their domains;
- (ii) there exist constants $C_1, C_2, C_3 > 0$ independent of u such that

$$\left| \int_{\Omega} (\alpha \nabla u)^T \nabla \bar{u} + \int_{\Gamma_V} p_2 (\alpha \nabla_S u)^T \nabla_S \bar{u} + C_2 \int_{\Omega} |u|^2 + C_3 \int_{\Gamma_V \cup \Gamma_R} |u|^2 \right| \geq C_1 \|u\|_V^2;$$

- (iii) $a(u, u) + b(u, u) = 0$ and $g_D \equiv 0$ together imply $u \equiv 0$;

- (iv) $f \in V', g_D \in H_{00}^{\frac{1}{2}}(\Gamma_D), g_R \in H^{-\frac{1}{2}}(\Gamma_R), g_V \in H^{-\frac{1}{2}}(\Gamma_V)$.

We refer the reader to [164] for the definitions of the function spaces and the well-posedness, which is based on the Lax–Milgram lemma and the Fredholm alternative.

4. Direct and Iterative Solvers. After discretization of (11), we obtain the linear system

$$(17) \quad A\mathbf{u} = \mathbf{f},$$

where A is an N -by- N matrix, \mathbf{u} is the solution to be sought, and \mathbf{f} is given.

Gaussian elimination is probably the oldest⁷ and the most fundamental solver for linear algebraic systems like (17). Its modern form consists in first computing the LU factorization $A = RPLUQ$ with R a row scaling diagonal matrix, L (U) being lower (upper) triangular, and P (Q) being row (column) permutations, and then solving $L\mathbf{v} = P(R^{-1}\mathbf{f})$ and $U(Q\mathbf{u}) = \mathbf{v}$. For a dense matrix A , e.g., from boundary element or spectral discretizations, the factorization requires $\mathcal{O}(N^3)$ flops and $\mathcal{O}(N^2)$ storage, and the triangular solves cost $\mathcal{O}(N^2)$ flops. For a sparse matrix A , e.g., from a low-order finite element discretization, one can benefit from the nonzero structure of the entries of A by ordering the eliminations such that as few nonzeros as possible are generated in L and U , because the number of nonzeros determines the storage and time complexities of the triangular solves; see, e.g., [39, 47]. Doing so on two-dimensional regular meshes, we need $\mathcal{O}(N^{\frac{3}{2}})$ flops for a sparse factorization, and the L, U factors one obtains have $\mathcal{O}(N \log N)$ nonzeros; see [86, 104]. On three-dimensional regular meshes, we need $\mathcal{O}(N^2)$ flops and get $\mathcal{O}(N^{4/3})$ nonzeros in L, U ; see [39, p. 143]. Significant progress has been made on reducing the actual constants hidden in the asymptotic complexities which is nowadays coded in widely used software packages like UMFPACK [38], PARDISO [150], SuperLU [117], and MUMPS [2]. The classical, factorization-based direct solvers aim at the exact solution and introduce no error in the algorithms other than round-off errors due to finite precision arithmetic. They have been proved in practice to be robust for various problems, and they are robust with respect to problem parameters. Moreover, they are very advantageous for multiple r.h.s., because the factorization can be reused for different r.h.s. just performing triangular solves, which is much faster than the factorization stage.

The drawbacks of direct solvers are the superlinear complexities they have in time and storage requirements, and also the important communication overhead in a parallel environment in both the factorization and the triangular solution stages.

A recent trend of direct solvers is their introducing low rank truncation of some off-diagonal dense blocks arising in the process of factorization. This is accomplished by \mathcal{H} -matrix techniques [8, 9, 10, 11, 93, 94] and related approaches [22, 88, 103, 126, 127, 170, 171, 172, 173, 175]. The numerical low rank property depends on the Green's function of the underlying PDE. In particular, for the Helmholtz equation, the numerical rank can be shown to grow in specific geometric conditions in two dimensions only logarithmically with the wavenumber (see [51, 127]). In general, however, as indicated in [9, p. 157], the growth seems to be linear in the wavenumber. In [95], the author says that there are two types of off-diagonal blocks: one type is small and can be treated easily by \mathcal{H} -matrix arithmetic, and the other type is large and is better treated by a multipole expansion; see [8] for more details. Recently, [54] gave lower and upper bounds for the separability of Green's function. Nearly linear complexities for two-dimensional Helmholtz problems have been presented in, e.g., [8, 88, 103, 126, 127, 170, 175], with fixed or increasing wavenumbers. For three-dimensional Helmholtz problems, although considerable improvements over classical direct solvers have been made using \mathcal{H} -matrix techniques, the numerical experiments

⁷Gaussian elimination can even be found in ancient Chinese scripts [122].

in [171, 173] show that the time complexity of factorization tends to $\mathcal{O}(N^{5/3}) \sim \mathcal{O}(N^2)$ at high wavenumber on proportionally refined meshes.

In contrast to direct solvers which deliver very accurate solutions in a finite number of operations, iterative solvers start from an initial guess and improve the accuracy successively by iteration. Iterative solvers have become a core area of research in numerical analysis.⁸ The central issue in designing an iterative solver is finding an approximation of A^{-1} which is called *preconditioner*. In this sense, the direct solvers with low rank truncation mentioned in the previous paragraph can be used as preconditioners; see, e.g., [8, 51, 175]. The simplest way to use a preconditioner $M^{-1} \approx A^{-1}$ is iterative refinement, also called Richardson iteration⁹ or deferred correction: $\mathbf{u} \leftarrow \mathbf{u} + M^{-1}(\mathbf{f} - \mathbf{A}\mathbf{u})$. More advanced are Krylov subspace methods; see [91, 145] for general introductions to iterative methods for linear systems. It is particularly difficult to design a fast iterative solver for the Helmholtz equation; see, e.g., the review papers [3, 55, 59]. The main challenge is to accomplish $\mathcal{O}(N)$ time complexity for increasing frequency ω of (11) on appropriately refined meshes; an easier goal is the linear complexity under mesh refinement for fixed frequency, because this does not add more propagating waves to the solution. To tackle the special difficulties of the Helmholtz equation, many techniques have been developed and integrated into three major frameworks: incomplete factorizations, (algebraic) multigrid, and domain decomposition. We will now briefly review some of these techniques.

The shifted-Laplace preconditioner M_ϵ^{-1} , proposed in [58, 109], introduces an imaginary shift ϵ to the frequency ω or ω^2 in the continuous problem given in (11), and M_ϵ is obtained from a discretization of the shifted operator. The analyses in [32, 37, 59, 66, 167] collectively exhibit a gap between the requirements that the shifted operator is close to the original operator and yet cheap to solve. In practice, $\mathcal{O}(\omega^2)$ imaginary shifts to ω^2 are often used. In this case, it is easy to find an iterative solver $\widetilde{M}_\epsilon^{-1}$ of $\mathcal{O}(N)$ complexity for M_ϵ for any ω ; but M_ϵ deviates from the original matrix A more and more as ω increases, so that the iteration numbers for the original system with the preconditioner $\widetilde{M}_\epsilon^{-1}$ also grow with ω and can be $\mathcal{O}(\omega)$ for the truncated free space problem or even $\mathcal{O}(\omega^2)$ in the case of a waveguide [32]. In the former case, we observed from numerical experiments in [14, 26, 36, 57, 90] that for two-dimensional problems with $N = \mathcal{O}(\omega^2)$ the overall time complexity becomes $\mathcal{O}(N^{3/2})$, and in [20, 36, 144] that for three-dimensional problems with $N = \mathcal{O}(\omega^3)$ it seems to be $\mathcal{O}(N^{4/3})$. Even though not optimal, for three-dimensional models and $\mathcal{O}(1)$ r.h.s. these iterative solvers can be faster than a complete factorization.

To accelerate convergence of iterations, an important idea is identifying the slowly convergent components of the errors and the corresponding residuals, and projecting them out from the iterates by solving the original problem restricted to the corresponding subspace. This is called coarse correction or deflation. When these slowly convergent coarse components are, however, based on a grid discretization, then for the Helmholtz equation the coarse problem for projection needs to be *fine* enough, typically of dimension $\mathcal{O}(\omega^2)$ in two dimensions and $\mathcal{O}(\omega^3)$ in three dimensions, to keep the convergence independent of ω ; see, e.g., [19, 116] for convergence theory with such rich coarse problems. This excessive requirement manifests inadequacy of the basic iterations presumed in $\mathcal{O}(N)$ time for oscillatory waves. Of course, the more time we

⁸Trefethen [165]: “The name of the new game is *iteration with preconditioning*. Increasingly often it is not optimal to try to solve a problem exactly in one pass; instead, solve it approximately, then iterate.”

⁹The method Richardson proposed is much more sophisticated, including a relaxation parameter that changes with each iteration and is chosen to lead to an optimized polynomial [143].

allocate to the basic underlying iteration, the smaller the coarse problem will be that we need to compensate for inadequacies of the basic iteration. Another approach is to try to develop an efficient solver for the coarse problem. For the shifted-Laplace preconditioner, multilevel Krylov with multigrid deflation was studied in [56, 152], and it was seen that the first coarse level (with mesh size twice as coarse as the finest level) needs to be solved more and more accurately to keep iteration numbers from growing as ω increases. Another direction is to seek more efficient coarse problems. Those based on wave-ray or plane waves that originated from [16, 161] in the context of multigrid have become popular in algebraic multigrid methods (see, e.g., [136, 168]) and in domain decomposition methods (see, e.g., [40, 61, 81, 105]). Some recent developments include the bootstrap trick [123] to discover the slowly convergent subspace, the local eigenspaces [35] for heterogeneous media, and improvement in stability of the coarse problems [24, 159]. Complex-symmetric least squares formulations [89, 123, 136], numerical-asymptotic hybridization [138], and block Krylov methods for multiple r.h.s. [114] have also given further insight in the search for a scalable Helmholtz solver.

When applying domain decomposition methods (see, e.g., [42, 63, 115, 141, 153, 164]) to the Helmholtz equation, an immediate obstacle is the use of the usual Dirichlet or Neumann boundary conditions on subdomain interfaces. In particular, taking a diagonal block of A , one cannot ensure that it is nonsingular. For example, in the unit square $(0, 1)^2$ with homogeneous Dirichlet boundary conditions, the negative Laplace operator $-\Delta = -\partial_{xx} - \partial_{yy}$ has eigenvalues $(n^2 + m^2)\pi^2$ ($n, m = 1, 2, \dots$), so the Helmholtz operator $-\Delta - k^2$ is singular if k^2 is equal to one of these eigenvalues. This will not happen if a subdomain is sufficiently small, because the minimal eigenvalue will then be larger than the given $k^2 \in \mathbb{R}$: for example, in the square $(0, H)^2$, the minimal eigenvalue is $2\pi^2/H^2$, which can be made bigger than any given real number k^2 when H is sufficiently small. This strategy was adopted in [19, 61, 116], but the coarse problems then become very large to maintain scalability with so many subdomains. Another natural treatment is regularization of subdomain problems with ABCs (or layers), which helps verify assumption (iii) in section 3. For example, letting $\Delta u + k^2 u = 0$ in Ω ($k \in \mathbb{R}$) and $\partial_{\mathbf{n}} u + pu = 0$ on $\partial\Omega$, we have $\int_{\Omega} |\nabla u|^2 - k^2 |u|^2 + \int_{\partial\Omega} p |u|^2 = 0$; if $\text{Im } p \neq 0$, we obtain $\partial_{\mathbf{n}} u = u = 0$ on $\partial\Omega$ which implies $u \equiv 0$ in Ω by the unique continuation property (see [174]). The well-posedness of high-order ABCs can be found in, e.g., [166]. Regularization with a zeroth-order absorbing condition was first used by B. Després in his thesis [41] and later in, e.g., [17, 40, 62]. See [128] for a parallel numerical study with application to scattering problems. Actually, one gets even more from this choice: compared to the classical Schwarz method that uses Dirichlet transmission conditions, faster convergence was observed. This can be understood in the ideal case with *transparent* transmission conditions as we have seen in section 2 for the optimal Schwarz method [53, 133] motivated by the numerical study in [98]. Based on this principle, optimized Schwarz methods (see, e.g., [64] for an introduction, [67, 124, 140] for analyses, and [79, 80] for geometry-dependent optimization) leverage various approximations, i.e., absorbing transmission conditions or PMLs for fast convergence. For Helmholtz problems, the second-order Taylor expansion was used in [43], square-root based nonlocal conditions were studied in [34, 87], best approximations of zero and second order were sought in [30, 69, 76, 83, 84, 160], Padé approximants with complex-shifted wavenumbers were used in [15], PMLs were first employed in [148, 163], and recently some rational interpolants were tested in [110] for waveguide problems. For a numerical comparison of low-order and high-order transmission conditions for the overlapping Schwarz methods, we refer to [84].

Parallel to the development of optimized Schwarz methods, absorbing transmission conditions have also found use in the *analytic incomplete LU* (AILU) preconditioner; see [77, 78]. The idea is based on the identification of the DtN-based transparent transmission condition with the Schur complements arising in the block LU factorization we saw in section 2. An important improvement to the AILU preconditioner was made by the independent development in [52] using PML instead of the second-order approximation used in [78]. This triggered more studies on exploiting PML for the iterative solution of the Helmholtz equation in a forward and backward sweeping fashion; see, e.g., [28, 29, 60, 111, 139, 157, 169, 179, 180]. A recursive version that solves the two-dimensional subproblems in a three-dimensional domain recursively by decomposing them into one-dimensional lines and sweeping can be found in [120]; see also [178] for a similar idea. A recursive sweeping algorithm with low-order ABCs was proposed earlier [1]. Another double sweep process that extends the Dirichlet–Neumann alternating method [141] to many subdomains is proposed in [25]. In all these sweeping methods, there is little parallelism across the block solves, since the blocks (or subdomains) are passed through one by one, but in return an expensive coarse problem for connecting the blocks is avoided. Hence, the parallelism and the complexity within each block become crucial. In [139] for three-dimensional models, each block is kept quasi-two-dimensional with fixed thickness, and a tailored parallel direct solver for the quasi-two-dimensional problems is used. The sequential complexity was shown to be $\mathcal{O}(\delta^2 N^{4/3})$ for the setup and $\mathcal{O}(\delta N \log N)$ for the solve, where $\delta = \delta(k)$ is the thickness of the discrete PML on one side of each block. Instead of PML, hierarchical matrix approximations can also be used; see [6, 51]. More recently, in an effort to parallelize the sweeping preconditioner, the authors of [119] proposed decomposing the source term into subdomains and then simulating its influence on the other subdomains by sweeping from that subdomain toward the first and the last subdomains. The final approximation is then obtained by adding the solutions corresponding to the different subdomain sources. The sweeping methods have also been combined with a two-level method in [158], and with the sparsifying preconditioner [176] for a volume integral reformulation of the Helmholtz equation in [121, 181].

The methods above based on approximation of transparent boundary conditions are currently among the most promising iterative methods for the Helmholtz equation and more general wave propagation phenomena. In the following sections, we will explain how these methods were invented while following various paths from very different starting points, and we give a formulation of each method in a common notation that allows us to prove that each of these methods is in fact a special optimized Schwarz method distinct only in transmission conditions, overlaps, and/or implementation. A first such relation between the source transfer method and an optimized Schwarz method was discovered in the proceedings paper [27], and further relations were pointed out in [82].

5. Notation. To make the analogy we have seen between the block LU factorization and the optimal Schwarz algorithm mathematically rigorous, and then to show precisely how all the new Helmholtz solvers are related to one another, requires a common notation that works for all formulations. This formulation must permit the simultaneous use of overlapping and nonoverlapping blocks or subdomains, Green’s function formulations and volume discretizations, and very general transmission conditions including ABCs and PMLs, and all this for both continuous and discrete formulations. We introduce the reader to this notation in this section, as we intro-

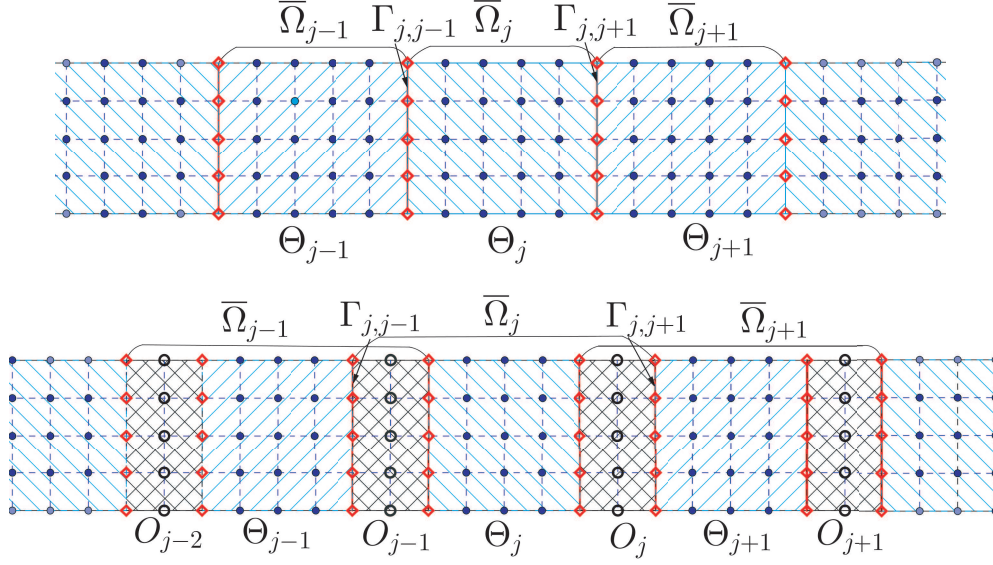


Fig. 2 Nonoverlapping and overlapping domain decomposition, $\bullet \in \Theta_*$, $\circ \in O_*$, $\diamond \in \Gamma_{*,\#}$.

duce the fundamental concepts common to all algorithms step by step. The first steps in learning the notation will be hard, but it will be rewarding to be able to understand the details of all these new Helmholtz solvers and their tight relationships.

5.1. Domain Decomposition. As we have seen, the algorithms are based on a decomposition, and we introduce this decomposition for the original domain Ω on which (11) is posed. We decompose $\bar{\Omega}$ into serially connected subdomains $\bar{\Omega}_j, j = 1, \dots, J$, such that

$$(18) \quad \bar{\Omega} = \cup_{j=1}^J \bar{\Omega}_j, \quad \bar{\Omega}_j \cap \bar{\Omega}_l = \emptyset \text{ if } |j - l| > 1.$$

To simplify the notation for the algorithms, we also introduce at each end an empty subdomain, $\Omega_0 = \Omega_{J+1} = \emptyset$. We denote the overlap between the subdomains by $O_j := \Omega_j \cap \Omega_{j+1}, j = 1, \dots, J-1$, the interfaces by $\Gamma_{j,j\pm 1} := \partial\Omega_j \cap \partial(\Omega_{j\pm 1} - \Omega_j), j, j\pm 1 \in \{1, \dots, J\}$, and the nonoverlapping region within each subdomain by $\Theta_j := \bar{\Omega}_j - (\bar{\Omega}_{j-1} \cup \bar{\Omega}_{j+1}), j = 1, \dots, J$, as indicated in Figure 2. Note that this partition can be considered either for the continuous domain or for the discretized domain. At the discrete level, the domain decomposition is a partition of the d.o.f. \mathbf{u} of (17). Corresponding to that partition, we introduce the notion of index sets as shown in Table 1, which we chose to make it easier to remember them: for example, in the nonoverlapping case, we have for subdomain Ω_j on the left boundary $\mathbf{u}_{j\langle} = \mathbf{u}_j = \mathbf{u}_{j\rfloor}$, and similarly on the right boundary $\mathbf{u}_{j\rangle} = \mathbf{u}_{j\rfloor} = \mathbf{u}_{j\rfloor}$, and the unknowns interior to subdomain Ω_j are $\mathbf{u}_{j\circ} = \mathbf{u}_{j\#}$, where a dot always means only the interior unknowns, and we have equality because the overlap is empty. As another example, if $\Theta_j \neq \emptyset$ in the overlapping case, we have $\mathbf{u}_{j\circ} = [\mathbf{u}_{j\#}; \mathbf{u}_j; \mathbf{u}_{j\#}; \mathbf{u}_{j\#}; \mathbf{u}_{j\#}]$.¹⁰ We denote by $I_*^\#$ the 0-1 restriction or extension matrix from the set of d.o.f. indicated by $\#$ to the

¹⁰We use MATLAB notation for concatenating column vectors vertically to avoid having to use the transpose symbol T .

Table I *Meaning of the subscripted vector \mathbf{v}_* .*

Subscript *	Meaning of the vector \mathbf{v}_*
$j\langle$ or j	$\bar{\Omega}_j$: all the d.o.f. on the j th subdomain $\bar{\Omega}_j$
$j\langle$	left interface of Ω_j : restriction of \mathbf{v}_j to the interface $\Gamma_{j,j-1}$
$j\rangle$	right interface of Ω_j : restriction of \mathbf{v}_j to the interface $\Gamma_{j,j+1}$
$j\langle\langle$	left overlap of Ω_j : restriction of \mathbf{v}_j to the overlap $\bar{\Omega}_j \cap \bar{\Omega}_{j-1}$
$j\rangle\rangle$	right overlap of Ω_j : restriction of \mathbf{v}_j to the overlap $\bar{\Omega}_j \cap \bar{\Omega}_{j+1}$
$j]$	end of left overlap in Ω_j : restriction of \mathbf{v}_j to $\Gamma_{j-1,j}$
$j[$	end of right overlap in Ω_j : restriction of \mathbf{v}_j to $\Gamma_{j+1,j}$
$j\langle\bullet$ or $j\bullet$	interior unknowns of Ω_j : after removing $\mathbf{v}_{j\langle}$ and $\mathbf{v}_{j\rangle}$ from $\mathbf{v}_{j\langle\langle}$
$j\langle\langle\langle$	interior of left overlap of Ω_j : after removing $\mathbf{v}_{j\langle}$ and $\mathbf{v}_{j]}$ from $\mathbf{v}_{j\langle\langle}$
$j\rangle\rangle\rangle$	interior of right overlap of Ω_j : after removing $\mathbf{v}_{j[}$ and $\mathbf{v}_{j\rangle}$ from $\mathbf{v}_{j\rangle\rangle}$
$j\langle\langle\langle\langle$	interior without overlaps of Ω_j : after removing $\mathbf{v}_{j\langle\langle}$ and $\mathbf{v}_{j\rangle\rangle}$ from $\mathbf{v}_{j\langle\langle}$
$\sim j$	everything to the left of Ω_j , i.e., all the d.o.f. in $(\cup_{m=1}^{j-1} \bar{\Omega}_m) - \bar{\Omega}_j$
$j\sim$	everything to the right of Ω_j , i.e., all the d.o.f. in $(\cup_{m=j+1}^J \bar{\Omega}_m) - \bar{\Omega}_j$

set indicated by *. For example, $I_{j\langle}^{j-1}$ is the restriction from $\bar{\Omega}_{j-1}$ to $\Gamma_{j,j-1}$. For the identity matrix I_* we simply write I_* . A common restriction operator we will often use is $R_* := I_*^{1,\dots,J}$. Note that by \mathbf{v}_l and \mathbf{v}_j we do not mean in general the restriction $\mathbf{v}_l = R_l \mathbf{v}$ and $\mathbf{v}_j = R_j \mathbf{v}$ of a common \mathbf{v} , but that the components of \mathbf{v}_j correspond to the d.o.f. on $\bar{\Omega}_j$. Hence, $I_{j\langle}^j \mathbf{v}_j$ can be different from $I_{j+1\langle}^{j+1} \mathbf{v}_{j+1}$. Only for the exact solution \mathbf{u} and the r.h.s. \mathbf{f} of (17) do we use \mathbf{u}_j and \mathbf{f}_j for $R_j \mathbf{u}$ and $R_j \mathbf{f}$. For submatrices of A , we also use the subscripts $j[$ interchangeably with $j+1\langle$ because they refer to the same index set in the global case, and similarly for $j]$ and $j-1\rangle$.

If $\Theta_j \neq \emptyset$ for $j = 1, \dots, J$, we assume that (17) has block tridiagonal structure¹¹ when the vector of unknowns is partitioned as $\mathbf{u} = [\mathbf{u}_{1\langle}; \mathbf{u}_{1\langle\langle}; \dots; \mathbf{u}_{J-1\rangle}; \mathbf{u}_{J\langle}],$ i.e.,

$$(19) \quad \begin{bmatrix} A_{1\langle} & A_{1\langle\langle} & & & & \\ A_{1\langle\langle} & A_{1\langle\langle\langle} & A_{1\langle\langle\langle\langle} & & & \\ & \ddots & \ddots & \ddots & & \\ & & A_{J-1\rangle\rangle} & A_{J-1\rangle} & A_{J-1\rangle J\langle} & \\ & & & A_{J\langle\langle} & A_{J\langle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1\langle} \\ \mathbf{u}_{1\langle\langle} \\ \vdots \\ \mathbf{u}_{J-1\rangle} \\ \mathbf{u}_{J\langle} \end{bmatrix} = \mathbf{f}.$$

For overlapping decompositions (i.e., $O_j \neq \emptyset$), we can also partition the overlap, $\mathbf{u}_{j\langle\langle} = [\mathbf{u}_{j[}; \mathbf{u}_{j\langle\langle}; \mathbf{u}_{j\rangle}]$, and similar to the block tridiagonal assumption, we assume that there are no direct interactions between d.o.f. across the interfaces, e.g.,

$$A_{j\langle\langle j\langle} = 0, \quad A_{j\rangle j\langle} = 0, \quad A_{j[j+1\langle} = 0, \quad A_{j\langle\langle j+1\langle} = 0.$$

In the nonoverlapping case we also use $j\rangle$ for $j\rangle\rangle$, since the index sets \rangle and $\rangle\rangle$ coincide in that case, and the same holds for \langle and $\langle\langle$.

¹¹This holds naturally for classical finite difference and finite element discretizations which approximate derivatives by only looking at neighboring nodes.

Remark 5. When $\Theta_1, \Theta_J \neq \emptyset$, $\Theta_j = \emptyset$ for $j = 2, \dots, J-1$, and $O_j \neq \emptyset$ for $j = 1, \dots, J-1$, we assume (17) is block tridiagonal under the partition

$$\mathbf{u} = [\mathbf{u}_{1\mathbb{H}}; \mathbf{u}_{1\mathbb{I}}; \mathbf{u}_{1\mathbb{D}}; \mathbf{u}_1; \mathbf{u}_{2\mathbb{D}}; \mathbf{u}_2; \dots; \mathbf{u}_{J-1\mathbb{I}}; \mathbf{u}_{J\mathbb{H}}].$$

Then all our subsequent discussions on (19) can be adapted to this case without essential difference. This case corresponds to what is called *generous overlap* in domain decomposition, i.e., almost every subdomain consists of overlaps with neighbors: $\Omega_j = O_{j-1} \cup \Gamma_{j-1,j} \cup O_j$ and $\Gamma_{j+1,j} = \Gamma_{j-1,j}$ for $j = 2, \dots, J-1$. For convenience, we will also denote $O_0 := \Theta_1$ and $O_J := \Theta_J$ for later use in subsection 8.1.

Remark 6. The algorithms we will discuss below also permit $A_{j\mathbb{D}l\mathbb{D}} \neq 0$ for $|j-l| = 1$, which can happen, for example, for spectral element discretizations with only one spectral element over the entire subdomain width. However, we exclude this situation for simplicity in what follows.

Remark 7. The block tridiagonal form we assumed in (19) is natural if the d.o.f. in the overlap \overline{O}_j including its boundaries are the same and shared by Ω_j and Ω_{j+1} , i.e., the problem stems from a globally assembled problem. In domain decomposition, however, nonmatching grids are also common, and we may have two sets of d.o.f. in \overline{O}_j , one set $\mathbf{u}_{j\mathbb{D}}$ for Ω_j and another set $\mathbf{u}_{j+1\mathbb{I}}$ for Ω_{j+1} . In this case, when $O_j \neq \emptyset$ we may assume (17) has the form of the *augmented system*

$$\begin{bmatrix} A_{1\mathbb{H}} & A_{1\mathbb{H}1\mathbb{D}} & & & & \\ A_{1\mathbb{D}1\mathbb{H}} & A_{1\mathbb{D}} & A_{1\mathbb{D}2\mathbb{I}} & \boxed{A_{1\mathbb{D}2\mathbb{H}}} & & \\ \boxed{A_{2\mathbb{I}1\mathbb{H}}} & A_{2\mathbb{I}1\mathbb{D}} & A_{2\mathbb{I}} & A_{2\mathbb{I}2\mathbb{H}} & & \\ & & \ddots & \ddots & \ddots & \boxed{A_{J-1\mathbb{D}J\mathbb{H}}} \\ & \boxed{A_{J\mathbb{I}J-1\mathbb{H}}} & A_{J\mathbb{I}J-1\mathbb{D}} & A_{J\mathbb{I}} & A_{J\mathbb{I}J\mathbb{H}} & \\ & & & A_{J\mathbb{H}J\mathbb{I}} & A_{J\mathbb{H}} & \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1\mathbb{H}} \\ \mathbf{u}_{1\mathbb{D}} \\ \mathbf{u}_{2\mathbb{I}} \\ \vdots \\ \mathbf{u}_{J\mathbb{I}} \\ \mathbf{u}_{J\mathbb{H}} \end{bmatrix} = \mathbf{f},$$

which would be block tridiagonal if we removed the boxed blocks. This form also arises naturally from nonconforming discretizations in the overlaps and on the interfaces, e.g., from certain discontinuous Galerkin methods (see, e.g., [4]). It is possible to generalize our discussions to this case, but we would not gain more insight and will thus not do so here to avoid further complications in the notation.

5.2. Transmission Conditions. We now present the three fundamental types of transmission conditions used by the algorithms: Dirichlet, Neumann, and generalized Robin conditions. For the Dirichlet condition, based on the assumptions in subsection 5.1, if we take the rows for $\mathbf{u}_{j\bullet}$ from (17), we find $A_{j\bullet}\mathbf{u}_{j\bullet} = \mathbf{f}_{j\bullet} - A_{j\bullet j\langle}\mathbf{u}_{j\langle} - A_{j\bullet j\langle}\mathbf{u}_{j\langle}$. We rewrite this as a linear system for $\mathbf{u}_{j\langle}$ with the interface data $\mathbf{u}_{j\langle}$ and $\mathbf{u}_{j\langle}$ provided by the neighboring subdomains,

$$(20) \quad \begin{bmatrix} I_{j\langle} & & \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\langle} \\ & I_{j\langle} & \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\bullet} \\ \mathbf{u}_{j\langle} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{j-1\mathbb{I}} \\ \mathbf{f}_{j\bullet} \\ \mathbf{u}_{j+1\mathbb{I}} \end{bmatrix}.$$

At the continuous level, this corresponds to (11) localized to Ω_j with Dirichlet transmission conditions on the interfaces,

$$(21) \quad \begin{aligned} \mathcal{L} u_j &= f && \text{in } \Omega_j, \\ \mathcal{B} u_j &= g && \text{on } \partial\Omega_j \cap \partial\Omega, \\ u_j &= u_{j-1} && \text{on } \Gamma_{j,j-1}, \\ u_j &= u_{j+1} && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

where $u_l := u|_{\Omega_l}$ ($l = j, j \pm 1$). As mentioned before, the subdomain problem in (21) might not be well-posed if we replace u_j by v_j and then try to solve (21) for v_j . Similarly, the matrix in (20) might not be invertible.

For the Neumann condition, we split the diagonal block of A corresponding to the interface $\Gamma_{j,j-1}$ into two parts as would arise naturally from the splitting of the bilinear forms in (16) in a conforming finite element method: $A_{j\langle} = A_{j\langle}^{\rangle} + A_{j\langle}^{\langle}$, where $A_{j\langle}^{\rangle}$ is the contribution from the left side of $\Gamma_{j,j-1}$ and $A_{j\langle}^{\langle}$ is the contribution from the right side of $\Gamma_{j,j-1}$. The reader might wonder why we use the superscript \rangle here (\rangle for the contribution from the left and the superscript \langle for the contribution from the right. The reason is that the contribution from the right will be used on the left of the corresponding subdomain that uses it, and vice versa, and based on the assumptions in subsection 5.1, the rows from (17) for \mathbf{u}_j can then be written similar to the Dirichlet case in the form

$$(22) \quad \begin{bmatrix} A_{j\langle}^{\langle} & A_{j\langle j\bullet} \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\langle} \\ & A_{j\langle j\bullet} & A_{j\langle}^{\rangle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\bullet} \\ \mathbf{u}_{j\langle} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j\langle} - A_{j\langle j-1\langle} \mathbf{u}_{j-1\langle} - A_{j\langle}^{\rangle} \mathbf{u}_{j-1\langle} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\langle} - A_{j\langle j+1\langle} \mathbf{u}_{j+1\langle} - A_{j\langle}^{\langle} \mathbf{u}_{j+1\langle} \end{bmatrix},$$

and now the superscript looks very natural and easy to remember. Equation (22) corresponds to (11) localized to Ω_j with Neumann transmission conditions on the interfaces,

$$(23) \quad \begin{aligned} \mathcal{L} u_j &= f && \text{in } \Omega_j, \\ \mathcal{B} u_j &= g && \text{on } \partial\Omega_j \cap \partial\Omega, \\ \mathbf{n}_j^T(\alpha \nabla u_j) &= \mathbf{n}_j^T(\alpha \nabla u_{j-1}) && \text{on } \Gamma_{j,j-1}, \\ \mathbf{n}_j^T(\alpha \nabla u_j) &= \mathbf{n}_j^T(\alpha \nabla u_{j+1}) && \text{on } \Gamma_{j,j+1}. \end{aligned}$$

In particular, we note that the discretization of $-\mathbf{n}_j^T(\alpha \nabla u_j)$ on $\Gamma_{j,j-1}$ gives $\mathbf{f}_{j\langle}^{\langle} - A_{j\langle}^{\langle} \mathbf{u}_{j\langle} - A_{j\langle j\bullet} \mathbf{u}_{j\bullet}$ and the discretization of $\mathbf{n}_j^T(\alpha \nabla u_{j-1})$ on $\Gamma_{j,j-1}$ gives $\mathbf{f}_{j\langle}^{\rangle} - A_{j\langle}^{\rangle} \mathbf{u}_{j-1\langle} - A_{j\langle j-1\langle} \mathbf{u}_{j-1\langle}$, where $\mathbf{f}_{j\langle} = \mathbf{f}_{j\langle}^{\rangle} + \mathbf{f}_{j\langle}^{\langle}$ is again the splitting of the contribution from the two sides of $\Gamma_{j,j-1}$. Note that as in the case of Dirichlet conditions, if we replace u_j by v_j in (23), the resulting subdomain problem might not be well-posed, and the matrix in (22) might not be invertible.

By generalized Robin transmission conditions, we mean the generalized linear combination of Dirichlet and Neumann conditions, i.e.,

$$(24) \quad \mathcal{Q}(\mathbf{n}_j^T(\alpha \nabla u_j)) + \mathcal{P} u_j = \mathcal{Q}(\mathbf{n}_j^T(\alpha \nabla u_l)) + \mathcal{P} u_l \text{ on } \Gamma_{j,l},$$

where \mathcal{Q} and \mathcal{P} are linear, possibly nonlocal operators along $\Gamma_{j,l}$, $l = j \pm 1$. At the discrete level, this corresponds to a generalized linear combination of the interface

rows of (20) and (22), while the interior rows are unchanged, i.e.,

$$\begin{aligned}
 (25) \quad & \begin{bmatrix} Q_{j\langle}^{\langle} A_{j\langle}^{\langle} + P_{j\langle}^{\langle} & Q_{j\langle}^{\langle} A_{j\langle j\bullet} & \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\langle} \\ & Q_{j\langle}^{\langle} A_{j\langle j\bullet} & Q_{j\langle}^{\langle} A_{j\langle}^{\langle} + P_{j\langle}^{\langle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\bullet} \\ \mathbf{u}_{j\langle} \end{bmatrix} \\
 &= \begin{bmatrix} Q_{j\langle}^{\langle} (\mathbf{f}_{j\langle} - A_{j\langle j-1\mathbb{H}} \mathbf{u}_{j-1\mathbb{H}}) + (P_{j\langle}^{\langle} - Q_{j\langle}^{\langle} A_{j\langle}^{\langle}) \mathbf{u}_{j-1\mathbb{H}} \\ \mathbf{f}_{j\bullet} \\ Q_{j\langle}^{\langle} (\mathbf{f}_{j\langle} - A_{j\langle j+1\mathbb{H}} \mathbf{u}_{j+1\mathbb{H}}) + (P_{j\langle}^{\langle} - Q_{j\langle}^{\langle} A_{j\langle}^{\langle}) \mathbf{u}_{j+1\mathbb{H}} \end{bmatrix}.
 \end{aligned}$$

If $Q_{j\langle}^{\langle} = I_{j\langle}$ and $Q_{j\langle}^{\langle} = I_{j\langle}$, we can also rewrite (25) without the explicit splitting of $A_{j\langle}$ and $A_{j\langle}$,

$$(26) \quad \begin{bmatrix} \tilde{S}_{j\langle}^{\langle} & A_{j\langle j\bullet} \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\langle} \\ & A_{j\langle j\bullet} & \tilde{S}_{j\langle}^{\langle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\bullet} \\ \mathbf{u}_{j\langle} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j\langle} - A_{j\langle j-1\mathbb{H}} \mathbf{u}_{j-1\mathbb{H}} + (\tilde{S}_{j\langle}^{\langle} - A_{j\langle}) \mathbf{u}_{j-1\mathbb{H}} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\langle} - A_{j\langle j+1\mathbb{H}} \mathbf{u}_{j+1\mathbb{H}} + (\tilde{S}_{j\langle}^{\langle} - A_{j\langle}) \mathbf{u}_{j+1\mathbb{H}} \end{bmatrix},$$

where $\tilde{S}_{j\langle}^{\langle} = A_{j\langle}^{\langle} + P_{j\langle}^{\langle}$ and $\tilde{S}_{j\langle}^{\langle} = A_{j\langle}^{\langle} + P_{j\langle}^{\langle}$. If we first specify $\tilde{S}_{j\langle}^{\langle}$ and $\tilde{S}_{j\langle}^{\langle}$, then it is not necessary to introduce a splitting of $A_{j\langle}$ and $A_{j\langle}$ to use (26).

We now consider a special case of (11): we assume that the data f and g are supported only on $\bar{\Omega}_j$ and vanish elsewhere. Suppose we are interested in the solution of (11) in Ω_j only. Then it would be desirable to have a problem equivalent to the original problem in (11) but defined just on the truncated domain Ω_j . This can be done by setting a *transparent boundary condition* on the truncation boundary $\partial\Omega_j - \partial\Omega$ and solving

$$\begin{aligned}
 (27) \quad & \mathcal{L} u_j = f \quad \text{in } \Omega_j, \\
 & \mathcal{B} u_j = g \quad \text{on } \partial\Omega_j \cap \partial\Omega, \\
 & \mathbf{n}_j^T (\alpha \nabla u_j) + \text{DtN}_j u_j = 0 \quad \text{on } \partial\Omega_j - \partial\Omega,
 \end{aligned}$$

where DtN_j is a linear operator defined as follows.

DEFINITION 8. *The Dirichlet-to-Neumann (DtN) operator exterior to $\Omega_j \subset \Omega$ for (11) is*

$$\begin{aligned}
 (28) \quad & \text{DtN}_j : d \rightarrow -\mathbf{n}_j^T (\alpha \nabla v), \text{ s.t. } \quad \mathcal{L} v = 0 \quad \text{in } \Omega - \Omega_j, \\
 & \quad \mathcal{B} v = 0 \quad \text{on } \partial\Omega - \partial\Omega_j, \\
 & \quad v = d \quad \text{on } \partial\Omega_j - \partial\Omega,
 \end{aligned}$$

where \mathbf{n}_j is the unit outward normal vector of Ω_j .

REMARK 9. The DtN operator is an example of a Poincaré–Steklov operator referring to maps between different boundary data. According to [108, p. VI], this class of operators was first studied by V. A. Steklov [156] and H. Poincaré [137]. They are also related to the Calderon projectors; see, e.g., [134].

LEMMA 10. *Assume that f and g in (11) vanish outside $\bar{\Omega}_j$. If (28) has a unique solution $v \in H^1(\Omega - \Omega_j)$ for $d \in H^{1/2}(\partial\Omega_j - \partial\Omega)$ and (11) has a unique solution $u \in H^1(\Omega)$, then (27) also has a unique solution $u_j \in H^1(\Omega_j)$, and $u_j = u|_{\Omega_j}$.*

Proof. In Definition 8, we substitute $u_j|_{\partial\Omega_j-\partial\Omega}$ for d and combine it with (27) to find

$$(29) \quad \begin{aligned} \mathcal{L}u_j &= f && \text{in } \Omega_j, && \mathcal{L}v &= 0 && \text{in } \Omega - \Omega_j, \\ \mathcal{B}u_j &= g && \text{on } \partial\Omega_j \cap \partial\Omega, && \mathcal{B}v &= 0 && \text{on } \partial\Omega - \partial\Omega_j, \\ \mathbf{n}_j^T(\alpha\nabla u_j) &= \mathbf{n}_j^T(\alpha\nabla v) && \text{on } \partial\Omega_j - \partial\Omega, && v &= u_j && \text{on } \partial\Omega_j - \partial\Omega. \end{aligned}$$

This coupled system for (u_j, v) has at least one solution $(u|_{\Omega_j}, u|_{\Omega-\Omega_j})$. For uniqueness, we set $f := 0$ and $g := 0$ and show that $u_j \equiv 0$ if $u_j \in H^1(\Omega_j)$: similar to (16), we test the PDE satisfied by u_j and v separately with arbitrary $w \in V$,

$$\begin{aligned} a_j(u_j, w) + b_j(u_j, w) &= \int_{\partial\Omega_j-\partial\Omega} \mathbf{n}_j^T(\alpha\nabla u_j) \bar{w}, \\ a_j^c(v, w) + b_j^c(v, w) &= \int_{\partial\Omega_j-\partial\Omega} -\mathbf{n}_j^T(\alpha\nabla v) \bar{w}, \end{aligned}$$

where the bilinear forms a_j and b_j correspond to a and b in (16) with the integration domains restricted to $\bar{\Omega}_j$, and similarly for a_j^c and b_j^c with the corresponding restriction to $\bar{\Omega} - \Omega_j$. Adding the two equations above and using the Neumann transmission condition from (29), the integral terms cancel and we obtain

$$a_j(u_j, w) + a_j^c(v, w) + b_j(u_j, w) + b_j^c(v, w) = 0.$$

Now we need to recombine these bilinear forms into the entire forms defined on the original function space on Ω . Given $u_j \in H^1(\Omega_j)$ and $u_j|_{\partial\Omega_j-\partial\Omega} \in H^{1/2}$, we have $v \in H^1(\Omega - \Omega_j)$ by assumption. If we define $\tilde{u} := u_j$ in Ω_j and $\tilde{u} := v$ in $\Omega - \Omega_j$, we know that $\tilde{u} \in H^1(\Omega)$ because $u_j = v$ on $\partial\Omega_j - \partial\Omega$ from (29). Hence, we have found a $\tilde{u} \in H^1(\Omega)$ satisfying $a(\tilde{u}, w) + b(\tilde{u}, w) = 0$. By uniqueness of the solution of (11), we then conclude that $\tilde{u} \equiv 0$. \square

The matrix analogue of the exact truncation is simply Gaussian elimination: if we consider (19) with \mathbf{f} nonzero *only* in \mathbf{f}_j , then we can rewrite (19) as

$$(30) \quad \begin{bmatrix} A_{\sim j} & A_{\sim j, j\langle} & & & \\ A_{j\langle, \sim j} & A_{j\langle} & A_{j\langle, j\bullet} & & \\ & A_{j\bullet, j\langle} & A_{j\bullet} & A_{j\bullet, j\langle} & \\ & & A_{j\langle, j\bullet} & A_{j\langle} & A_{j\langle, j\sim} \\ & & & A_{j\sim, j\langle} & A_{j\sim} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\sim j} \\ \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\bullet} \\ \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\sim} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{f}_{j\langle} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\langle} \\ 0 \end{bmatrix}.$$

To obtain the truncated model for \mathbf{u}_j only, we eliminate $\mathbf{u}_{\sim j}$ and $\mathbf{u}_{j\sim}$ and find

$$(31) \quad \begin{bmatrix} S_{j\langle}^{\langle} & A_{j\langle, j\bullet} \\ A_{j\bullet, j\langle} & A_{j\bullet} & A_{j\bullet, j\langle} \\ & A_{j\langle, j\bullet} & S_{j\langle}^{\langle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\bullet} \\ \mathbf{u}_{j\langle} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j\langle} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\langle} \end{bmatrix},$$

where $S_{j\langle}^{\langle} = A_{j\langle} - A_{j\langle, \sim j} A_{\sim j}^{-1} A_{\sim j, j\langle}$ and $S_{j\langle}^{\langle} = A_{j\langle} - A_{j\langle, j\sim} A_{j\sim}^{-1} A_{j\sim, j\langle}$ are known as Schur complements, which are usually dense matrices. Similar to Lemma 10, we have the following lemma.

LEMMA 11. *If in (30) the coefficient matrix and the diagonal blocks $A_{\sim j}$, $A_{j\sim}$ are invertible, then (31) is uniquely solvable and its solution is part of the solution of (30).*

Remark 12. If $j = 1$, then there is no $\mathbf{u}_{j\langle}$, and similarly if $j = J$, then there is no $\mathbf{u}_{j\rangle}$, so the corresponding rows and columns in (30) and (31) should be deleted. In this case, Lemma 11 still holds. From now on, we will treat $j = 1$ and $j = J$ like the other j 's and just assume that the nonexistent blocks are deleted.

Recalling the splitting $A_{j\langle} = A_{j\langle}^{\rangle} + A_{j\langle}^{\langle}$, we can interpret (31) as containing generalized Robin boundary conditions similar to (25) by writing $S_{j\langle}^{\langle} = A_{j\langle}^{\langle} + P_{j\langle}^{\langle}$ and $S_{j\rangle}^{\rangle} = A_{j\rangle}^{\rangle} + P_{j\rangle}^{\rangle}$ with

$$(32) \quad P_{j\langle}^{\langle} := A_{j\langle}^{\rangle} - A_{j\langle, \sim j} A_{\sim j}^{-1} A_{\sim j, j\langle}, \quad P_{j\rangle}^{\rangle} := A_{j\rangle}^{\langle} - A_{j\rangle, \sim j} A_{\sim j}^{-1} A_{\sim j, j\rangle}.$$

Comparing (27) and (31), both for the exact truncation, we find $P_{j\langle}^{\langle}$ should be the analogue of the Dirichlet-to-Neumann operator DtN_j restricted to its input argument on $\Gamma_{j,j-1}$. We can also observe directly from (32) that $P_{j\langle}^{\langle}$ acts on Dirichlet data $\mathbf{v}_{j\langle}$ by $-A_{\sim j, j\langle}$ (negative sign for moving to the r.h.s.), solves for $\mathbf{v}_{\sim j}$ in the exterior by $A_{\sim j}^{-1}$ and then evaluates the Neumann data by $A_{j\langle}^{\rangle} \mathbf{v}_{j\langle} + A_{j\langle, \sim j} \mathbf{v}_{\sim j}$.

Remark 13. PML is a popular technique to approximate the transparent boundary condition, and it is appropriate to make a connection here between the practical implementation of PML and our present discussion. The PML technique replaces the original problem exterior to Ω_j by a modified one on another exterior domain Ω_j^{pml} surrounding Ω_j along the truncation boundary $\partial\Omega_j - \partial\Omega$. A DtN operator for the modified problem in Ω_j^{pml} can be defined as¹²

$$(33) \quad \begin{aligned} \text{DtN}_j^{pml} : (d_1, d_2) &\rightarrow -\mathbf{n}_j^T(\tilde{\alpha}\nabla v) && \text{on } (\Gamma_{j,j-1}, \Gamma_{j,j+1}), \\ \text{s.t. } \tilde{\mathcal{L}}v &= 0 && \text{in } \Omega_j^{pml}, \\ \tilde{\mathcal{B}}v &= 0 && \text{on } \partial\Omega_j^{pml} - \partial\Omega_j, \\ v &= d_1 && \text{on } \Gamma_{j,j-1} \subset (\partial\Omega_j^{pml} \cap \partial\Omega_j), \\ v &= d_2 && \text{on } \Gamma_{j,j+1} \subset (\partial\Omega_j^{pml} \cap \partial\Omega_j), \end{aligned}$$

where $\tilde{\mathcal{L}} := -\nabla^T \tilde{\alpha} \nabla - \omega^2/\tilde{\kappa}$. Then an approximate transparent boundary condition can be used in the case stated in Lemma 10 to obtain $\tilde{u}_j \approx u_j$:

$$(34) \quad \begin{aligned} \mathcal{L} \tilde{u}_j &= f && \text{in } \Omega_j, \\ \mathcal{B} \tilde{u}_j &= g && \text{on } \partial\Omega_j \cap \partial\Omega, \\ \mathbf{n}_j^T(\alpha \nabla \tilde{u}_j) + \text{DtN}_j^{pml} \tilde{u}_j &= 0 && \text{on } \partial\Omega_j - \partial\Omega. \end{aligned}$$

To actually solve (34), as we did in the proof of Lemma 10, we substitute with (33) and compose a problem defined on $\tilde{\Omega}_j := \bar{\Omega}_j \cup \bar{\Omega}_j^{pml}$,

$$\begin{aligned} \tilde{\mathcal{L}} \tilde{u}_j &= \tilde{f} && \text{in } \tilde{\Omega}_j, \\ \tilde{\mathcal{B}} \tilde{u}_j &= \tilde{g} && \text{on } \partial\tilde{\Omega}_j, \end{aligned}$$

where $\tilde{\mathcal{L}} = \mathcal{L}$ in Ω_j , $\tilde{\mathcal{B}} = \mathcal{B}$ on $\partial\Omega_j \cap \partial\Omega$, and \tilde{f}, \tilde{g} are the zero extensions of f, g .

¹²In our setting, except for $j = 1, J$, Ω_j^{pml} has two disconnected parts, one on the left side of $\Gamma_{j,j-1}$ and one on the right side of $\Gamma_{j,j+1}$. So DtN_j^{pml} is block diagonal in the sense that $\mathbf{n}_j^T(\tilde{\alpha}\nabla v)$ on $\Gamma_{j,j-1}$ depends only on d_1 , and on $\Gamma_{j,j+1}$ it depends only on d_2 .

Remark 14. At the matrix level, the PML technique corresponds to replacing (30) with

$$(35) \quad \begin{bmatrix} \tilde{A}_{\sim j} & \tilde{A}_{\sim j, j\langle} & & & \\ \tilde{A}_{j\langle, \sim j} & \tilde{A}_{j\langle} & A_{j\langle j\bullet} & & \\ & A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\langle} & \\ & & A_{j\langle j\bullet} & \tilde{A}_{j\langle} & \tilde{A}_{j\langle, j\sim} \\ & & & \tilde{A}_{j\sim, j\langle} & \tilde{A}_{j\sim} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_{\sim j} \\ \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\bullet} \\ \mathbf{u}_{j\langle} \\ \tilde{\mathbf{u}}_{j\sim} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{f}_{j\langle} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\langle} \\ 0 \end{bmatrix},$$

where the entries with tildes (except $\tilde{A}_{j\langle}$ and $\tilde{A}_{j\langle}$) are typically of much smaller dimension than the original ones. The Schur complemented system of (35), intended to approximate (31), is

$$(36) \quad \begin{bmatrix} \tilde{S}_{j\langle}^{\langle} & A_{j\langle j\bullet} \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\langle} \\ & A_{j\langle j\bullet} & \tilde{S}_{j\langle}^{\langle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle} \\ \mathbf{u}_{j\bullet} \\ \mathbf{u}_{j\langle} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j\langle} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\langle} \end{bmatrix},$$

where $\tilde{S}_{j\langle}^{\langle} := \tilde{A}_{j\langle} - \tilde{A}_{j\langle, j\sim} \tilde{A}_{j\sim}^{-1} \tilde{A}_{j\sim, j\langle}$ and $\tilde{S}_{j\langle}^{\rangle} := \tilde{A}_{j\langle} - \tilde{A}_{j\langle, j\sim} \tilde{A}_{j\sim}^{-1} \tilde{A}_{j\sim, j\langle}$. As before, we see that $\tilde{P}_{j\langle}^{\langle} := \tilde{S}_{j\langle}^{\langle} - A_{j\langle}^{\langle}$ is the matrix version of the PML-DtN operator $\text{DtN}_j^{\text{pml}}$ restricted to its input argument on $\Gamma_{j, j-1}$. For implementation, one usually does not solve (36) directly, but instead solves (35).

5.3. Green's Function. So far we have prepared the reader for the new Helmholtz solvers based only on the concepts of domain decomposition and transmission conditions. However, there are also formulations of these new Helmholtz solvers based on Green's functions, which we introduce next. By definition, a fundamental solution $G(\mathbf{x}, \mathbf{y})$ of the partial differential operator \mathcal{M} is a solution of the PDE in a domain Y without consideration of boundary conditions,

$$\mathcal{M}_{\mathbf{y}} G(\mathbf{x}, \mathbf{y}) = \delta(\mathbf{y} - \mathbf{x}) \quad \forall \mathbf{x} \in X \subseteq Y,$$

where $\mathcal{M}_{\mathbf{y}}$ is the operator \mathcal{M} acting on the \mathbf{y} variable and $\delta(\mathbf{y} - \mathbf{x})$ is the Dirac delta function representing a point source and satisfying $\int_Y \delta(\mathbf{y} - \mathbf{x}) v(\mathbf{y}) d\mathbf{y} = v(\mathbf{x})$. Let u be a solution of $\mathcal{L}u = f$ in Y (see (11) without boundary condition), and $\mathcal{M} := -\nabla^T(\alpha^T \nabla \cdot) - \frac{\omega^2}{\kappa}$. Using integration by parts, we have formally for $\mathbf{x} \in X - \partial Y$

$$(37) \quad u(\mathbf{x}) = \int_Y G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} + \int_{\partial Y} \mathbf{n}_{\mathbf{y}}^T (\alpha \nabla u(\mathbf{y})) G(\mathbf{x}, \mathbf{y}) - \mathbf{n}_{\mathbf{y}}^T (\alpha^T \nabla_{\mathbf{y}} G(\mathbf{x}, \mathbf{y})) u(\mathbf{y}) d\sigma(\mathbf{y}),$$

which is a representation formula for the solution, and the three summands are called volume potential, single layer potential, and double layer potential; see [134]. A justification of (37) involves existence, regularity, and singularity of the fundamental solution and the solution, which can be found in the literature if α , κ , f , and ∂X are bounded and smooth; see, e.g., [142, 147]. We note that the r.h.s. of (37) uses both Neumann and Dirichlet traces of u , while a well-posed boundary condition usually only tells us one of them or a generalized linear combination of them. For example, let $u = 0$ on ∂Y . To obtain a usable representation, we can require G to also satisfy $G(\mathbf{x}, \mathbf{y}) = 0$ for $\mathbf{y} \in \partial Y$, $\mathbf{x} \in X$. Then the single layer and the double layer potentials in (37) vanish, and we obtain the simple representation formula

$$(38) \quad u(\mathbf{x}) = \int_Y G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}.$$

We call a *fundamental solution* satisfying a homogeneous boundary condition *Green's function*. However, the two terms are sometimes used in an exchangeable way.

Remark 15. If u satisfies an inhomogeneous boundary condition $\mathcal{B}u|_{\partial Y} = g$, we can lift (extend) the boundary data into Y , i.e., find a function v on Y such that $\mathcal{B}v|_{\partial Y} = g$, and subtract it from u so that the boundary condition becomes homogeneous for the new unknown $\tilde{u} := u - v$. (We will see that this trick is useful also for the implementation of the transmission condition in (24).) For the Green's function G , we impose $\mathcal{B}_y^T G(\mathbf{x}, \mathbf{y}) = 0$, where \mathcal{B}^T corresponds to \mathcal{B} but with α replaced by α^T .

Remark 16. Another convention is to define the Green's function $G(\mathbf{x}, \mathbf{y})$ through $\mathcal{L}_x G(\mathbf{x}, \mathbf{y}) = \delta(\mathbf{x} - \mathbf{y})$ equipped with homogeneous boundary conditions as for u . Then (38) can be obtained by the superposition principle. Similarly, we may define $H(\mathbf{x}, \mathbf{y})$ through $\mathcal{L}_y H(\mathbf{x}, \mathbf{y}) = \delta(\mathbf{y} - \mathbf{x})$. We then have $H(\mathbf{y}, \mathbf{x}) = G(\mathbf{x}, \mathbf{y})$. Furthermore, if $\alpha = \alpha^T$, we have $G(\mathbf{x}, \mathbf{y}) = G(\mathbf{y}, \mathbf{x})$.

We now point out an analogy between (38) and the solution $\mathbf{u} = A^{-1}\mathbf{f}$ of (17). For a particular value of \mathbf{x} , $u(\mathbf{x})$ in (38) corresponds to a particular (say, the m th) entry of \mathbf{u} , and $G(\mathbf{x}, \mathbf{y})$ then also corresponds to the m th row of A^{-1} , and the integral in (38) becomes the inner product of the row of A^{-1} with \mathbf{f} . Similarly, for a particular \mathbf{y} , $G(\mathbf{x}, \mathbf{y})$ corresponds to a particular column of A^{-1} . We now take a closer look at the Schur complement $S_{j\langle}^{\langle}$ in (31), which is essentially derived from the 2-by-2 block matrix by Gaussian elimination,

$$(39) \quad \begin{bmatrix} A_{\sim j} & A_{\sim j, j\langle} \\ A_{j\langle, \sim j} & A_{j\langle} \end{bmatrix} = \begin{bmatrix} I_{\sim j} & \\ A_{j\langle, \sim j} A_{\sim j}^{-1} & I_{j\langle} \end{bmatrix} \begin{bmatrix} A_{\sim j} & A_{\sim j, j\langle} \\ S_{j\langle}^{\langle} \end{bmatrix}.$$

Taking the inverse of both sides, we find

$$= \begin{bmatrix} A_{\sim j} & A_{\sim j, j\langle} \\ A_{j\langle, \sim j} & A_{j\langle} \end{bmatrix}^{-1} = \begin{bmatrix} A_{\sim j} & A_{\sim j, j\langle} \\ S_{j\langle}^{\langle} \end{bmatrix}^{-1} \begin{bmatrix} I_{\sim j} & \\ -A_{j\langle, \sim j} A_{\sim j}^{-1} & I_{j\langle} \end{bmatrix} \\ = \begin{bmatrix} I_{\sim j} & -A_{\sim j}^{-1} A_{\sim j, j\langle} \\ & I_{j\langle}^{\langle} \end{bmatrix} \begin{bmatrix} A_{\sim j}^{-1} & \\ & S_{j\langle}^{\langle -1} \end{bmatrix} \begin{bmatrix} I_{\sim j} & \\ -A_{j\langle, \sim j} A_{\sim j}^{-1} & I_{j\langle} \end{bmatrix} = \begin{bmatrix} * & * \\ * & S_{j\langle}^{\langle -1} \end{bmatrix},$$

where we omit the terms marked by *. Recalling the analogy between the matrix inverse and the Green's function, we can identify $S_{j\langle}^{\langle -1}$ as a diagonal part of the Green's function $G(\mathbf{x}, \mathbf{y})$. Here, G satisfies for $\mathbf{x}, \mathbf{y} \in \Omega_{\sim j}^{+h}$

$$\begin{aligned} \mathcal{M}_y G(\mathbf{x}, \mathbf{y}) &= \delta(\mathbf{y} - \mathbf{x}) && \text{in } \Omega_{\sim j}^{+h}, \\ \mathcal{B}_y^T G(\mathbf{x}, \mathbf{y}) &= 0 && \text{on } \partial\Omega \cap \partial\Omega_{\sim j}^{+h}, \\ G(\mathbf{x}, \mathbf{y}) &= 0 && \text{on } \Gamma_{j, j-1}^{+h}, \end{aligned}$$

where $\Omega_{\sim j}^{+h}$ is the domain covering the left part of $\Omega - \Omega_j$ but with one grid layer further into Ω_j , and $\Gamma_{j, j-1}^{+h}$ is the interface of $\Omega_{\sim j}^{+h}$ in Ω_j . We see that $S_{j\langle}^{\langle -1}$ corresponds to $G(\mathbf{x}, \mathbf{y})$ with \mathbf{x}, \mathbf{y} both restricted to $\Gamma_{j, j-1}$. This identification was first given in [51, 52].

6. Optimized Schwarz Methods. If we have $u_j = u|_{\Omega_j}$, $j = 1, \dots, J$, with u the solution of (11), we must have consistency: (a) u_j and u_l matching¹³ on $\overline{\Omega_j} \cap \overline{\Omega_l}$, and (b) the original equations are satisfied in neighborhoods of interfaces $\partial\Omega_j \cap \overline{\Omega_l}$ for all $j \neq l$, $j, l \in \{1, \dots, J\}$. Conversely, if u_j , $j = 1, \dots, J$, solves (11) restricted to $\overline{\Omega_j}$, then (a) and (b) together imply $u_j = u|_{\Omega_j}$. For second-order elliptic PDEs like (11), (b) is equivalent to saying the Neumann traces $\mathbf{n}^T \alpha \nabla u_j$, $\mathbf{n}^T \alpha \nabla u_l$ match¹⁴ on $\partial\Omega_j \cap \overline{\Omega_l}$ for all $j, l \in \{1, \dots, J\}$. Hence, when $\{\Omega_j\}_{j=1}^J$ are nonoverlapping, (a) and (b) reduce to both Dirichlet and Neumann (or any other equivalent pair of) traces to match on every interface. If the subdomains $\{\Omega_j\}_{j=1}^J$ overlap, (a) and (b) as a whole can be further condensed as (a') *one* transmission condition (matching one of Dirichlet/Neumann/generalized Robin traces) on every interface, and (b') the transmission conditions ensure $u_j = u_l$ on $\overline{\Omega_j} \cap \overline{\Omega_l}$ if u_j and u_l both solve the original equations restricted to overlaps. Therefore, no matter whether the decomposition is overlapping or nonoverlapping, the original problem can be rewritten as a system of subdomain problems coupled through transmission conditions on interfaces; cf. [141].

Schwarz methods split the coupling between subdomains by taking the interface data from the already available iterates and solving subdomain problems to obtain the new iterate.¹⁵ Historically, the first Schwarz method was the *alternating Schwarz method* introduced by Schwarz himself [151], where one subdomain is solved at a time and then the newest data is passed on to the neighboring subdomains. This is analogous to the Gauss–Seidel iteration in linear algebra. More than a century later, Lions introduced the so-called *parallel Schwarz method* [118], where each subdomain solves its local problem at the same time and data is only exchanged afterward. This is analogous to the Jacobi iteration in linear algebra. In the alternating Schwarz method in the presence of many subdomains, one also needs to specify an ordering, and for the Helmholtz solvers we are interested in here with the decomposition into a one-dimensional sequence of subdomains, the particular ordering of sweeping from the first subdomain to the last one and then back, like in the symmetrized Gauss–Seidel iteration in linear algebra, is important, and we call these “double sweep” methods.

One also has to decide upon which unknowns are used to write the iteration: one can choose subdomain approximations (for the equivalent coupled system), global approximations (for the original problem), interface data, and residuals. We explain now in detail these formulations and their relations.

6.1. Subdomain Transmission Form of Schwarz Methods. In this formulation, the iterates represent approximate solutions on the subdomains. The corresponding *double sweep optimized Schwarz method* (DOSM) was first proposed in [129, 130]. Based on the decomposition defined in (18), we state the DOSM in Algorithm 1 at the PDE level for (11) and in Algorithm 2 at the matrix level for (19). Note that the transmission conditions on the interfaces can be changed in the process, e.g., from the forward sweep to the backward sweep or from one iteration to the next. Note

¹³Usually, “match” means “coincide.” But there are exceptions; e.g., the original problem can enforce a jump of the solution across a surface, or at the discrete level nonconforming discretization is used in overlaps and interfaces.

¹⁴We assume all surface/line/point sources on the interface have been split and incorporated into subdomain problems.

¹⁵Our description is also applicable to single-trace methods such as BDD, FETI, and FETI-H, which are based on nonoverlapping decompositions and use the same data for neighboring subdomains on each interface. In contrast, for Schwarz methods each subdomain is equipped with its own interface data which is provided by (but not used by) the other subdomains.

Algorithm 1. DOSM in the **subdomain transmission** form at the **PDE** level.

Given the last iterate $\{u_j^{(n-1)} \text{ in } \Omega_j, j = 1, \dots, J\}$, solve successively for $j = 1, \dots, J-1$,

$$\begin{aligned} \mathcal{L} u_j^{(n-\frac{1}{2})} &= f && \text{in } \Omega_j, \\ \mathcal{B} u_j^{(n-\frac{1}{2})} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathcal{Q}_j^{\langle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^{\langle} u_j^{(n-\frac{1}{2})} &= \mathcal{Q}_j^{\langle} \left(\mathbf{n}_j^T \alpha \nabla u_{j-1}^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^{\langle} u_{j-1}^{(n-\frac{1}{2})} && \text{on } \Gamma_{j,j-1}, \\ \mathcal{Q}_j^{\rangle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^{\rangle} u_j^{(n-\frac{1}{2})} &= \mathcal{Q}_j^{\rangle} \left(\mathbf{n}_j^T \alpha \nabla u_{j+1}^{(n-1)} \right) + \mathcal{P}_j^{\rangle} u_{j+1}^{(n-1)} && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

where \mathcal{Q}_j^{\langle} , \mathcal{P}_j^{\langle} and \mathcal{Q}_j^{\rangle} , \mathcal{P}_j^{\rangle} are some possibly nonlocal operators on the interfaces $\Gamma_{j,j-1}$ and $\Gamma_{j,j+1}$, and \mathbf{n}_j is the unit outward normal vector of Ω_j . We call this process the forward sweep.

Then the backward sweep consists in solving successively for $j = J, \dots, 1$,

$$\begin{aligned} \mathcal{L} u_j^{(n)} &= f && \text{in } \Omega_j, \\ \mathcal{B} u_j^{(n)} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathcal{Q}_j^{\langle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n)} \right) + \mathcal{P}_j^{\langle} u_j^{(n)} &= \mathcal{Q}_j^{\langle} \left(\mathbf{n}_j^T \alpha \nabla u_{j-1}^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^{\langle} u_{j-1}^{(n-\frac{1}{2})} && \text{on } \Gamma_{j,j-1}, \\ \mathcal{Q}_j^{\rangle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n)} \right) + \mathcal{P}_j^{\rangle} u_j^{(n)} &= \mathcal{Q}_j^{\rangle} \left(\mathbf{n}_j^T \alpha \nabla u_{j+1}^{(n)} \right) + \mathcal{P}_j^{\rangle} u_{j+1}^{(n)} && \text{on } \Gamma_{j,j+1}. \end{aligned}$$

Algorithm 2. DOSM in the **subdomain transmission** form at the **matrix** level.

Given the last iterate $\{\mathbf{u}_j^{(n-1)}, j = 1, \dots, J\}$, solve successively for $j = 1, \dots, J-1$,

$$= \begin{bmatrix} \mathcal{Q}_{j\langle}^{\langle} A_{j\langle}^{\langle} + P_{j\langle}^{\langle} & \mathcal{Q}_{j\langle}^{\langle} A_{j\langle}^{\bullet} & & \\ & A_{j\bullet\langle} & A_{j\bullet} & A_{j\bullet\jmath} \\ & & \mathcal{Q}_{j\jmath}^{\rangle} A_{j\jmath}^{\bullet} & \mathcal{Q}_{j\jmath}^{\rangle} A_{j\jmath}^{\langle} + P_{j\jmath}^{\rangle} \\ & & & \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle}^{(n-\frac{1}{2})} \\ \mathbf{u}_{j\bullet}^{(n-\frac{1}{2})} \\ \mathbf{u}_{j\jmath}^{(n-\frac{1}{2})} \end{bmatrix} \\ = \begin{bmatrix} \mathcal{Q}_{j\langle}^{\langle} \left(\mathbf{f}_{j\langle} - A_{j\langle j-1\mathbb{H}} \mathbf{u}_{j-1\mathbb{H}}^{(n-\frac{1}{2})} \right) + \left(P_{j\langle}^{\langle} - \mathcal{Q}_{j\langle}^{\langle} A_{j\langle}^{\langle} \right) \mathbf{u}_{j-1\mathbb{I}}^{(n-\frac{1}{2})} \\ \mathbf{f}_{j\bullet} \\ \mathcal{Q}_{j\jmath}^{\rangle} \left(\mathbf{f}_{j\jmath} - A_{j\jmath j+1\mathbb{H}} \mathbf{u}_{j+1\mathbb{H}}^{(n-1)} \right) + \left(P_{j\jmath}^{\rangle} - \mathcal{Q}_{j\jmath}^{\rangle} A_{j\jmath}^{\langle} \right) \mathbf{u}_{j+1\mathbb{I}}^{(n-1)} \end{bmatrix},$$

which constitutes the forward sweep.

Then perform the backward sweep: solve successively for $j = J, \dots, 1$,

$$= \begin{bmatrix} \mathcal{Q}_{j\langle}^{\langle} A_{j\langle}^{\langle} + P_{j\langle}^{\langle} & \mathcal{Q}_{j\langle}^{\langle} A_{j\langle}^{\bullet} & & \\ & A_{j\bullet\langle} & A_{j\bullet} & A_{j\bullet\jmath} \\ & & \mathcal{Q}_{j\jmath}^{\rangle} A_{j\jmath}^{\bullet} & \mathcal{Q}_{j\jmath}^{\rangle} A_{j\jmath}^{\langle} + P_{j\jmath}^{\rangle} \\ & & & \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle}^{(n)} \\ \mathbf{u}_{j\bullet}^{(n)} \\ \mathbf{u}_{j\jmath}^{(n)} \end{bmatrix} \\ = \begin{bmatrix} \mathcal{Q}_{j\langle}^{\langle} \left(\mathbf{f}_{j\langle} - A_{j\langle j-1\mathbb{H}} \mathbf{u}_{j-1\mathbb{H}}^{(n-\frac{1}{2})} \right) + \left(P_{j\langle}^{\langle} - \mathcal{Q}_{j\langle}^{\langle} A_{j\langle}^{\langle} \right) \mathbf{u}_{j-1\mathbb{I}}^{(n-\frac{1}{2})} \\ \mathbf{f}_{j\bullet} \\ \mathcal{Q}_{j\jmath}^{\rangle} \left(\mathbf{f}_{j\jmath} - A_{j\jmath j+1\mathbb{H}} \mathbf{u}_{j+1\mathbb{H}}^{(n)} \right) + \left(P_{j\jmath}^{\rangle} - \mathcal{Q}_{j\jmath}^{\rangle} A_{j\jmath}^{\langle} \right) \mathbf{u}_{j+1\mathbb{I}}^{(n)} \end{bmatrix}.$$

If $\mathcal{Q}_{j\langle}^{\langle} = I$ and $\mathcal{Q}_{j\jmath}^{\rangle} = I$, one can rewrite the subproblems as in (26).

also that in the double sweep, the subproblem on the last subdomain Ω_J is solved only once. If the transmission conditions on $\Gamma_{1,2}$ are the same in the forward and the backward sweeps, we find the same problem on Ω_1 is solved in the backward sweep of the current iteration and in the forward sweep of the next iteration, so one can also solve it only once.

For the *parallel optimized Schwarz method* (POSM), where all subdomains are solved simultaneously and data is exchanged afterward, it was shown in [133] that if optimal transmission conditions based on the DtN operators are used, then the algorithm converges in a finite number of steps, equal to the number of subdomains, and thus the iteration operator is nilpotent of degree equal to the number of subdomains. We present now an optimal choice for the DOSM, where the operators \mathcal{Q}_j^\rangle and \mathcal{P}_j^\rangle can still be arbitrary, as long as the subdomain problems are well-posed.

THEOREM 17. *If in the forward and the backward sweeps \mathcal{Q}_j^\langle is the identity, $\mathcal{P}_j^\langle = \text{DtN}_j^\langle := \text{DtN}_j|_{\Gamma_{j,j-1}}$ is well-defined for $j = 2, \dots, J$ as in Definition 8, and the original problem in (11) and the subdomain problems in Algorithm 1 are uniquely solvable, then Algorithm 1 converges in one double sweep for an arbitrary initial guess, and $u_j^{(1)} = u|_{\Omega_j}$, $j = 1, \dots, J$, with u the solution of (11). This means that the iteration operator of the DOSM is nilpotent of degree one.*

Proof. We note that the subdomain problems in Algorithm 1 are satisfied by the solution u . By linearity, it is thus sufficient to prove $u_j^{(1)} \equiv 0$ when $f \equiv 0$ and $g \equiv 0$. We first consider $u_2^{(\frac{1}{2})}$, which satisfies the transmission condition

$$(40) \quad \mathbf{n}_2^T \alpha \nabla u_2^{(\frac{1}{2})} + \text{DtN}_2^\langle u_2^{(\frac{1}{2})} = \mathbf{n}_2^T \alpha \nabla u_1^{(\frac{1}{2})} + \text{DtN}_2^\langle u_1^{(\frac{1}{2})} \text{ on } \Gamma_{2,1}.$$

Since $\mathcal{L}u_1^{(\frac{1}{2})} = 0$ in $\Omega_1 \supset \Omega_{\sim 2}$ and $\mathcal{B}u_1^{(\frac{1}{2})} = 0$ on $(\partial\Omega \cap \partial\Omega_1) \supset (\partial\Omega \cap \partial\Omega_{\sim 2})$, from Definition 8 we have $\text{DtN}_2^\langle u_1^{(\frac{1}{2})} = -\mathbf{n}_2^T \alpha \nabla u_1^{(\frac{1}{2})}$. Substituting this into (40) we obtain

$$\mathbf{n}_2^T \alpha \nabla u_2^{(\frac{1}{2})} + \text{DtN}_2^\langle u_2^{(\frac{1}{2})} = 0 \text{ on } \Gamma_{2,1}.$$

Now assuming that

$$(41) \quad \mathbf{n}_j^T \alpha \nabla u_j^{(\frac{1}{2})} + \text{DtN}_j^\langle u_j^{(\frac{1}{2})} = 0 \text{ on } \Gamma_{j,j-1},$$

we will show that this also holds for $j+1$ instead of j . In fact, by the assumption that DtN_{j+1}^\langle is well-defined, we have a unique solution $v_{\sim j+1}$ of the problem

$$\begin{aligned} \mathcal{L}v_{\sim j+1} &= 0 && \text{in } \Omega_{\sim j+1}, \\ \mathcal{B}v_{\sim j+1} &= 0 && \text{on } \partial\Omega \cap \partial\Omega_{\sim j+1}, \\ v_{\sim j+1} &= u_j^{(\frac{1}{2})} && \text{on } \Gamma_{j+1,j}. \end{aligned}$$

By Lemma 10, we have from (41) that $u_j^{(\frac{1}{2})} = v_{\sim j+1}$ in $\Omega_j \cap \Omega_{\sim j+1}$. Therefore, $\text{DtN}_{j+1}^\langle u_j^{(\frac{1}{2})} = -\mathbf{n}_{j+1}^T \alpha \nabla v_{\sim j+1} = -\mathbf{n}_{j+1}^T \alpha \nabla u_j^{(\frac{1}{2})}$ on $\Gamma_{j+1,j}$. Substituting this into the transmission condition for $u_{j+1}^{(\frac{1}{2})}$ we find

$$\mathbf{n}_{j+1}^T \alpha \nabla u_{j+1}^{(\frac{1}{2})} + \text{DtN}_{j+1}^\langle u_{j+1}^{(\frac{1}{2})} = 0 \text{ on } \Gamma_{j+1,j}.$$

By induction, (41) holds for all $j = 2, \dots, J-1$ and also $j = J$ except that we write $u_j^{(1)}$ instead of $u_j^{(\frac{1}{2})}$. By Lemma 10, and recalling that $f \equiv 0$ and $g \equiv 0$, we obtain $u_j^{(1)} = u|_{\Omega_j} \equiv 0$. Now assuming that

$$(42) \quad u_{j+1}^{(1)} \equiv 0 \text{ in } \Omega_{j+1},$$

we have to show that $u_j \equiv 0$ in Ω_j . This follows directly from (42) and (41), which imply that all the data in the problem for $u_j^{(1)}$ vanish, and from the assumption that the subdomain problem is uniquely solvable. \square

We also have the equivalent result of convergence in one step for the discrete case.

THEOREM 18. *If in the forward and the backward sweeps $Q_{j\langle}^{\langle} = I_{j\langle}$, $P_{j\langle}^{\langle} = A_{j\langle}^{\langle} - A_{j\langle} \sim_j A_{\sim_j}^{-1} A_{\sim_j, j\langle}$ is well-defined for $j = 2, \dots, J$, and the original problem in (19) and the subdomain problems in Algorithm 2 are uniquely solvable, then Algorithm 2 converges in one step and $\mathbf{u}_j^{(1)} = R_j \mathbf{u}$ with \mathbf{u} the solution of (19). This means the iteration matrix of the DOSM is nilpotent of degree one.*

Proof. First, the subdomain problems are consistent, i.e., neglecting the iteration numbers and substituting $\mathbf{u}_j = R_j \mathbf{u}$, we find the equations are satisfied by \mathbf{u} . Hence, by considering the errors, we only need to show that $\mathbf{u}_j^{(1)} = 0$ if $\mathbf{f} = 0$. In the problem for $\mathbf{u}_2^{(\frac{1}{2})}$, the r.h.s. corresponding to $\Gamma_{2,1}$ becomes

$$-A_{2\langle 1\mathbb{H}} \mathbf{u}_{1\mathbb{H}}^{(\frac{1}{2})} - A_{2\langle 1\mathbb{H}} A_{1\mathbb{H}}^{-1} A_{1\mathbb{H} 2\langle} \mathbf{u}_{1\mathbb{H}}^{(\frac{1}{2})} = -A_{2\langle 1\mathbb{H}} \left(\mathbf{u}_{1\mathbb{H}}^{(\frac{1}{2})} - A_{1\mathbb{H}}^{-1} A_{1\mathbb{H} 2\langle} \mathbf{u}_{1\mathbb{H}}^{(\frac{1}{2})} \right) = 0,$$

since $\mathbf{u}_1^{(\frac{1}{2})}$ satisfies $A_{1\mathbb{H}} \mathbf{u}_{1\mathbb{H}}^{(\frac{1}{2})} + A_{1\mathbb{H} 2\langle} \mathbf{u}_{1\mathbb{H}}^{(\frac{1}{2})} = 0$. In other words, we have for $j = 2$,

$$(43) \quad S_{j\langle}^{\langle} \mathbf{u}_{j\langle}^{(\frac{1}{2})} + A_{j\langle j\bullet} \mathbf{u}_{j\bullet}^{(\frac{1}{2})} = 0.$$

Assuming (43) holds for one j in $\{2, \dots, J-1\}$, we will show it will also hold for $j+1$ instead of j . In the theorem we are proving, we have assumed that $A_{\sim j+1}$ is invertible, so we can introduce $\mathbf{v}_{\sim j+1} := [\mathbf{v}_{\sim j}; \mathbf{v}_{j-1\mathbb{D}}; \mathbf{v}_{j\mathbb{H}}]$ as the solution of

$$\begin{bmatrix} A_{\sim j} & A_{\sim j, j-1\mathbb{D}} \\ A_{j-1\mathbb{D} \sim j} & A_{j-1\mathbb{D}} & A_{j-1\mathbb{D} j\mathbb{H}} \\ & A_{j\mathbb{H} j-1\mathbb{D}} & A_{j\mathbb{H}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{\sim j} \\ \mathbf{v}_{j-1\mathbb{D}} \\ \mathbf{v}_{j\mathbb{H}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -A_{j\mathbb{H}, j+1\langle} \mathbf{u}_{j\mathbb{H}}^{(\frac{1}{2})} \end{bmatrix}.$$

By Gaussian elimination, we have

$$\begin{bmatrix} S_{j-1\mathbb{D}}^{\langle} & A_{j-1\mathbb{D} j\mathbb{H}} \\ A_{j\mathbb{H} j-1\mathbb{D}} & A_{j\mathbb{H}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{j-1\mathbb{D}} \\ \mathbf{v}_{j\mathbb{H}} \end{bmatrix} = \begin{bmatrix} 0 \\ -A_{\sim j+1, j+1\langle} \mathbf{u}_{j\mathbb{H}}^{(\frac{1}{2})} \end{bmatrix},$$

which is also satisfied by the restriction of $\mathbf{u}_j^{(\frac{1}{2})}$ because of (43), $\mathbf{f} \equiv 0$, and Algorithm 2. By Lemma 11, we have that $\mathbf{v}_{j-1\mathbb{D}} = \mathbf{u}_{j\mathbb{D}}^{(\frac{1}{2})}$ and $\mathbf{v}_{j\mathbb{H}} = \mathbf{u}_{j\mathbb{H}}^{(\frac{1}{2})}$. In Algorithm 2,

the r.h.s. for $\mathbf{u}_{j+1}^{(\frac{1}{2})}$ then becomes

$$\begin{aligned} & -A_{j+1\langle j} \mathbf{u}_{j\boxminus}^{(\frac{1}{2})} - A_{j+1\langle, \sim j+1} A_{\sim j+1}^{-1} A_{\sim j+1, j+1\langle} \mathbf{u}_{j\lceil}^{(\frac{1}{2})} \\ & = -A_{j+1\langle j} \mathbf{u}_{j\boxminus}^{(\frac{1}{2})} + A_{j+1\langle, \sim j+1} \mathbf{v}_{\sim j+1} \\ & = -A_{j+1\langle j} \mathbf{u}_{j\boxminus}^{(\frac{1}{2})} + A_{j+1\langle j} \mathbf{v}_{j\boxminus} \\ & = -A_{j+1\langle j} \mathbf{u}_{j\boxminus}^{(\frac{1}{2})} + A_{j+1\langle j} \mathbf{u}_{j\boxminus}^{(\frac{1}{2})} = 0. \end{aligned}$$

Now we know that (43) holds for $j = 2, \dots, J-1$ and $j = J$, except that we write $\mathbf{u}_J^{(1)}$ instead of $\mathbf{u}_J^{(\frac{1}{2})}$, and we see that the r.h.s. for $\mathbf{u}_J^{(1)}$ vanishes, so $\mathbf{u}_J^{(1)} = 0$. In the backward sweep, the r.h.s. on $\Gamma_{j,j-1}$ for $\mathbf{u}_j^{(1)}$ is by Algorithm 2, as it is for $\mathbf{u}_j^{(\frac{1}{2})}$, and hence is zero by (43), and the r.h.s. on $\Gamma_{j,j+1}$ vanishes, given $\mathbf{u}_{j+1}^{(1)} = 0$. By induction, we thus conclude that $\mathbf{u}_j^{(1)} = 0$, $j = 1, \dots, J$. \square

Algorithms 1 and 2 use the subdomain approximations as iterates. If we want to have a global approximation for the original problem as a final result, we can just glue together subdomain approximations after stopping the iteration. This can be done by setting $u^{(n)} := \sum_{j=1}^J \mathcal{E}_j(\phi_j u_j^{(n)})$ at the PDE level and $\mathbf{u}^{(n)} = \sum_{j=1}^J R_j^T \Phi_j \mathbf{u}_j^{(n)}$ at the matrix level, where \mathcal{E}_j is the extension by zero from Ω_j to Ω and ϕ_j is a weighting function and, correspondingly, Φ_j a diagonal matrix. For consistency, when $u_j^{(n)} = u|_{\Omega_j}$ with u the solution of the original problem, we want $u^{(n)} = u$, or $\sum_{j=1}^J \mathcal{E}_j(\phi_j u|_{\Omega_j}) = \sum_{j=1}^J (\mathcal{E}_j \phi_j) u = u$. To ensure this for arbitrary data of (11), we must have $\sum_{j=1}^J \mathcal{E}_j \phi_j \equiv 1$. At the matrix level, we must have $\sum_{j=1}^J R_j^T \Phi_j R_j = I$. In particular, for a nonoverlapping decomposition, we must have $\phi_j \equiv 1$ in Ω_j .

6.2. Global Deferred Correction Form of Schwarz Methods. If we want to use global approximations as iterates, i.e., input the last iterate $u^{(n-1)}$ to the DOSM and get $u^{(n)}$ as output, we need to be very careful with the weighting functions introduced in the last paragraph. This is because Algorithm 1 relies essentially on the interface data, and when inputting $u^{(n-1)}$ instead of $\{u_j^{(n-1)}\}$, we must ensure that the values of $\{u_j^{(n-1)}\}$ necessary for the evaluation of the interface data in Algorithm 1 can still be found in $u^{(n-1)}$. We thus need a variant of Algorithm 1 that generates the iterate $u^{(n)}$ without storing $u_j^{(n-1)}$ (just storing $u^{(n-1)}$), and satisfying $u^{(n)} = \sum_{j=1}^J \mathcal{E}_j \phi_j u_j^{(n)}$, given that this relation holds for $n = 0$. The result of this algorithm should be identical to the glued result from the iterates $\{u_j^{(n)}\}$ generated by Algorithm 1. The equivalence of the new variant to Algorithm 1, and Algorithm 2 at the discrete level, is important, because then Theorem 17, and Theorem 18 at the discrete level, will also hold for the variant. We present the DOSM version with global approximations as iterates in Algorithm 3 at the continuous level and in Algorithm 4 for the discrete case. For the parallel form of the algorithms, i.e., the POSM, the situation is even more delicate and has been studied at length in [48, 155]. For example, the well-known preconditioner [46], called the *additive Schwarz method* (AS), is designed to be symmetric but then loses any equivalence to the POSM. The AS also cannot be used as a standalone iterative method, since it is not convergent; for a discussion, see [48, 65].

Remark 19. Algorithms 3 and 4 require the weighting functions to take values either 0 or 1, and to constitute a partition of unity. The resulting way of gluing sub-

Algorithm 3. DOSM in the **global deferred correction** form at the **PDE** level.

Given the last iterate $u^{(n-1)}$, solve successively for $j = 1, \dots, J-1$,

$$\begin{aligned} \mathcal{L} v_j^{(n-\frac{1}{2})} &= f - \mathcal{L} u^{(n-1+\frac{j-1}{2J-1})} && \text{in } \Omega_j, \\ \mathcal{B} v_j^{(n-\frac{1}{2})} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathcal{Q}_j^\langle \left(\mathbf{n}_j^T \alpha \nabla v_j^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^\langle v_j^{(n-\frac{1}{2})} &= 0 && \text{on } \Gamma_{j,j-1}, \\ \mathcal{Q}_j^\rangle \left(\mathbf{n}_j^T \alpha \nabla v_j^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^\rangle v_j^{(n-\frac{1}{2})} &= 0 && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

with each solve followed by $u^{(n-1+\frac{j}{2J-1})} \leftarrow u^{(n-1+\frac{j-1}{2J-1})} + \mathcal{E}_j(\phi_j v_j^{(n-\frac{1}{2})})$ with $\mathcal{E}_j \phi_j$ forming a nonoverlapping partition of unity of Ω , i.e., $\phi_j = 1$ in its support contained in $\bar{\Omega}_j$, $(\mathcal{E}_j \phi_j)(\mathcal{E}_l \phi_l) \equiv 0$ for $j \neq l$ and $\sum_{j=1}^J \mathcal{E}_j \phi_j \equiv 1$ in Ω .

Then in the backward sweep solve successively for $j = J, \dots, 1$,

$$\begin{aligned} \mathcal{L} v_j^{(n)} &= f - \mathcal{L} u^{(n-1+\frac{2J-j-1}{2J-1})} && \text{in } \Omega_j, \\ \mathcal{B} v_j^{(n)} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathcal{Q}_j^\langle \left(\mathbf{n}_j^T \alpha \nabla v_j^{(n)} \right) + \mathcal{P}_j^\langle v_j^{(n)} &= 0 && \text{on } \Gamma_{j,j-1}, \\ \mathcal{Q}_j^\rangle \left(\mathbf{n}_j^T \alpha \nabla v_j^{(n)} \right) + \mathcal{P}_j^\rangle v_j^{(n)} &= 0 && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

with each solve followed by $u^{(n-1+\frac{2J-j}{2J-1})} \leftarrow u^{(n-1+\frac{2J-j-1}{2J-1})} + \mathcal{E}_j(\phi_j v_j^{(n)})$.

Algorithm 4. DOSM in the **global deferred correction** form at the **matrix** level.

Given the last iterate $\mathbf{u}^{(n-1)}$, solve successively for $j = 1, \dots, J-1$ (note that \mathbf{v}_j changes with n),

$$(44) \quad \begin{bmatrix} Q_{j\langle}^\langle A_{j\langle}^\langle + P_{j\langle}^\langle & Q_{j\langle}^\langle A_{j\langle j\bullet} & & \\ & A_{j\bullet} & A_{j\bullet j\bullet} & \\ A_{j\bullet j\langle} & & & \\ & Q_{j\langle}^\rangle A_{j\langle j\bullet} & Q_{j\langle}^\rangle A_{j\langle}^\rangle + P_{j\langle}^\rangle & \end{bmatrix} \begin{bmatrix} \mathbf{v}_{j\langle} \\ \mathbf{v}_{j\bullet} \\ \mathbf{v}_{j\langle} \end{bmatrix} = \begin{bmatrix} Q_{j\langle}^\langle \\ I_{j\bullet} \\ Q_{j\langle}^\rangle \end{bmatrix} R_j \left(\mathbf{f} - A \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} \right),$$

with each solve followed by

$$(45) \quad \mathbf{u}^{(n-1+\frac{j}{2J-1})} \leftarrow \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} + R_j^T \Phi_j \mathbf{v}_j$$

with the following constraints: Φ_j is a diagonal matrix with its diagonal containing ones on $\Theta_j \cup \Gamma_{j+1,j}$ and zeros on $\Gamma_{j,j-1}$, $(R_j^T \Phi_j) R_j (R_l^T \Phi_l) = 0$ as $j \neq l$, and $\sum_{j=1}^J R_j^T \Phi_j R_j = I_{1,\dots,J}$.

Then solve successively for $j = J, \dots, 1$,

$$\begin{bmatrix} Q_{j\langle}^\langle A_{j\langle}^\langle + P_{j\langle}^\langle & Q_{j\langle}^\langle A_{j\langle j\bullet} & & \\ & A_{j\bullet} & A_{j\bullet j\bullet} & \\ A_{j\bullet j\langle} & & & \\ & Q_{j\langle}^\rangle A_{j\langle j\bullet} & Q_{j\langle}^\rangle A_{j\langle}^\rangle + P_{j\langle}^\rangle & \end{bmatrix} \begin{bmatrix} \mathbf{v}_{j\langle} \\ \mathbf{v}_{j\bullet} \\ \mathbf{v}_{j\langle} \end{bmatrix} = \begin{bmatrix} Q_{j\langle}^\langle \\ I_{j\bullet} \\ Q_{j\langle}^\rangle \end{bmatrix} R_j \left(\mathbf{f} - A \mathbf{u}^{(n-1+\frac{2J-j-1}{2J-1})} \right),$$

with each solve followed by $\mathbf{u}^{(n-1+\frac{j}{2J-1})} \leftarrow \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} + R_j^T \Phi_j \mathbf{v}_j$. This time the diagonal matrix Φ_j has as its diagonal values ones on $\Theta_j \cup \Gamma_{j-1,j}$ and zeros on $\Gamma_{j,j+1}$, and the last two constraints are the same as before.

domain approximations into a global approximation was introduced in [18] under the name *restricted additive Schwarz method* (RAS), and this made the method equivalent to the underlying parallel Schwarz method, but at the price of sacrificing symmetry. The restricted weighting never adds two subdomain approximations at the same location so that the current subdomain approximation can be subtracted and updated through a correction. One can then use a global deferred correction at every substep. One could also evaluate the global residual at the beginning of the n th iteration and do the gluing and the global correction at the end of the n th iteration, while carrying out the intermediate substeps in the subdomain transmission form or a local deferred correction form (see subsection 8.1). Then, weighting functions from the backward sweep of Algorithm 4 without the restricted constraint $(R_j^T \Phi_j) R_j (R_l^T \Phi_l) = 0$ could be used for gluing the global correction.

Remark 20. Algorithm 3 (Algorithm 4) uses a restricted extension $\mathcal{E}_j \phi_j$ ($R_j^T \Phi_j$) of the local approximations, but a full restriction $\cdot|_{\Omega_j}$ (R_j) of the global residuals. A variant of each algorithm can be obtained by using a restricted restriction $\cdot|_{\Omega_j} \phi_j$ ($\Phi_j R_j$) of the global residuals but a full extension \mathcal{E}_j (R_j^T) of the local approximations. For example, in Algorithm 4, Φ_j could be moved from the right of R_j^T in (45) to the left of R_j in (44). This idea was first introduced in [18] to transform RAS into the *additive Schwarz method with harmonic extension* (ASH), and later adopted and studied in [113] for optimized Schwarz methods with overlap. In [113], a close relation was proved between the iterates of the optimized ASH and the POSM. Moreover, the harmonic extension variant of Algorithms 3 and 4 can be shown to converge in one step under the assumptions of Theorems 17 and 18. When the coefficient matrices of the original problem and the subproblems are complex symmetric (i.e., $A^T = A$), then the RAS and the ASH lead to preconditioned systems that are the transpose of each other, and hence they have the same spectra.

THEOREM 21. *Suppose the subproblems in Algorithm 3 are well-posed. For an overlapping decomposition, if ϕ_j of Algorithm 3 also satisfies $\mathbf{n}_j^T \alpha \nabla \phi_j = 0$ on $\Gamma_{j \pm 1, j}$, $j = 1, \dots, J$, then Algorithm 3 is equivalent to Algorithm 1. That is, given $u^{(0)} = \sum_{j=1}^J \mathcal{E}_j(\phi_j u_j^{(0)})$, the iterates generated by the two algorithms satisfy for all n that $u^{(n)} = \sum_{l=1}^J \mathcal{E}_l(\phi_l u_l^{(n)})$.*

Proof. Suppose $u^{(0)} = \sum_{j=1}^J \mathcal{E}_j(\phi_j u_j^{(0)})$ and the sequence $u^{(*)}$ is generated by Algorithm 3, and the sequence $u_j^{(*)}$ is generated by Algorithm 1. Assuming that $u^{(n-1+\frac{j-1}{2J-1})} = \sum_{l=1}^{j-1} \mathcal{E}_l(\phi_l u_l^{(n-\frac{1}{2})}) + \sum_{l=j}^J \mathcal{E}_l(\phi_l u_l^{(n-1)})$, we will show that the same relation also holds for $j+1$ instead of j . First, we have $u^{(n-1+\frac{j-1}{2J-1})} = u_{j-1}^{(n-\frac{1}{2})}$ on $\Gamma_{j,j-1}$ because $\phi_{j-1} = 1$ in Θ_{j-1} and ϕ_j is compactly supported in Ω_j . Second, we have $\mathbf{n}_j^T \alpha \nabla u^{(n-1+\frac{j-1}{2J-1})} = \mathbf{n}_j^T \alpha \nabla u_{j-1}^{(n-\frac{1}{2})}$ on $\Gamma_{j,j-1}$ because $\phi_{j-1} = 1$ in Θ_{j-1} , ϕ_j is compactly supported in Ω_j , ϕ_{j-1} is smooth, and $\mathbf{n}_j^T \alpha \nabla \phi_{j-1} = 0$ on $\Gamma_{j,j-1}$. Combining the two traces, we have that the transmission condition on $\Gamma_{j,j-1}$ in Algorithm 1 is equivalent to

$$\mathcal{Q}_j^{\langle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^{\langle} u_j^{(n-\frac{1}{2})} = \mathcal{Q}_j^{\langle} \left(\mathbf{n}_j^T \alpha \nabla u^{(n-1+\frac{j-1}{2J-1})} \right) + \mathcal{P}_j^{\langle} u^{(n-1+\frac{j-1}{2J-1})}.$$

By the same argument, the other transmission condition in the forward sweep can also be rewritten using $u^{(n-1+\frac{j-1}{2J-1})}$. We introduce $\tilde{v}_j := u_j^{(n-\frac{1}{2})} - u^{(n-1+\frac{j-1}{2J-1})}|_{\Omega_j}$ and we find from Algorithm 1 that \tilde{v}_j solves the forward sweeping problem of Algorithm 3.

Hence, $v_j^{(n-\frac{1}{2})} = \tilde{v}_j = u_j^{(n-\frac{1}{2})} - u^{(n-1+\frac{j-1}{2J-1})}|_{\Omega_j}$. By this equation, Algorithm 3, and our assumptions, we obtain

$$\begin{aligned} u^{(n-1+\frac{j-1}{2J-1})} &= u^{(n-1+\frac{j-1}{2J-1})} + \mathcal{E}_j(\phi_j v_j^{(n-\frac{1}{2})}) \\ &= (1 - \mathcal{E}_j \phi_j) u^{(n-1+\frac{j-1}{2J-1})} + \mathcal{E}_j(\phi_j u_j^{(n-\frac{1}{2})}) \\ &= \sum_{s \neq j} \mathcal{E}_s \phi_s \left(\sum_{l=1}^{j-1} \mathcal{E}_l(\phi_l u_l^{(n-\frac{1}{2})}) + \sum_{l=j}^J \mathcal{E}_l(\phi_l u_l^{(n-1)}) \right) + \mathcal{E}_j(\phi_j u_j^{(n-\frac{1}{2})}) \\ &= \sum_{l=1}^j \mathcal{E}_l(\phi_l u_l^{(n-\frac{1}{2})}) + \sum_{l=j+1}^J \mathcal{E}_l(\phi_l u_l^{(n-1)}). \end{aligned}$$

By induction, we know this holds for all $j = 1, \dots, J-1$. Similarly, we can prove for the backward sweep

$$u^{(n-1+\frac{2J-j}{2J-1})} = \sum_{l=1}^{j-1} \mathcal{E}_l(\phi_l u_l^{(n-\frac{1}{2})}) + \sum_{l=j}^J \mathcal{E}_l(\phi_l u_l^{(n)}) \text{ for all } j = J, \dots, 1.$$

In particular, $u^{(n)} = \sum_{l=1}^J \mathcal{E}_l(\phi_l u_l^{(n)})$, and the result follows by induction. \square

Remark 22. The assumption that $\mathbf{n}_j^T \alpha \nabla \phi_j = 0$ on $\Gamma_{j \pm 1, j}$ was first introduced in [28]. We think this assumption can be removed from Theorem 21. In the matrix version (see Theorem 23) we find no counterpart of this assumption. For the same reason, we think that Theorem 21 also holds for nonoverlapping decompositions. There are, however, some difficulties for the justification at the PDE level: as noted in [157], “ $u^{(*)}$ is generally discontinuous at the interface $\Gamma_{j, j-1}$, which results in a most singular residual, and the restriction of the residual to the subdomain Ω_j is not well defined.”

THEOREM 23. *If the subproblems in Algorithm 4 are well-posed, Algorithm 4 is equivalent to Algorithm 2. That is, given $\mathbf{u}^{(0)} = \sum_{j=1}^J R_j^T \Phi_j \mathbf{u}_j^{(0)}$, the iterates generated by the two algorithms satisfy $\mathbf{u}^{(n)} = \sum_{l=1}^J R_l^T \Phi_l \mathbf{u}_l^{(n)}$ for all n .*

Proof. Suppose $\mathbf{u}^{(0)} = \sum_{j=1}^J R_j^T \Phi_j \mathbf{u}_j^{(0)}$ and the iterates $\mathbf{u}^{(*)}$ are generated by Algorithm 4, and the iterates \mathbf{u}_j^* , $j = 1, \dots, J$, by Algorithm 2. Assuming that

$$(46) \quad \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} = \sum_{l=1}^{j-1} R_l^T \Phi_l \mathbf{u}_l^{(n-\frac{1}{2})} + \sum_{l=j}^J R_l^T \Phi_l \mathbf{u}_l^{(n-1)},$$

we will show that the same relation also holds for $j+1$ instead of j . By (46) and the constraints on Φ_l , we have

$$\begin{aligned} R_j \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} &= \mathbf{u}_{j-1\downarrow}^{(n-\frac{1}{2})}, & R_{j-1\downarrow} \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} &= \mathbf{u}_{j-1\downarrow}^{(n-\frac{1}{2})}, \\ R_j \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} &= \mathbf{u}_{j+1\uparrow}^{(n-1)}, & R_{j+1\uparrow} \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} &= \mathbf{u}_{j+1\uparrow}^{(n-1)}. \end{aligned}$$

Substituting these into the forward sweep problem in Algorithm 2, we find $\mathbf{u}_j^{(n-\frac{1}{2})} - R_j \mathbf{u}^{(n-1+\frac{j-1}{2J-1})}$ solves the forward sweep problem of Algorithm 4, i.e., $\mathbf{v}_j = \mathbf{u}_j^{(n-\frac{1}{2})} - R_j \mathbf{u}^{(n-1+\frac{j-1}{2J-1})}$. Substituting this into the update relation of Algorithm 4 and using

the constraints on Φ_l , we obtain

$$\begin{aligned} \mathbf{u}^{(n-1+\frac{j}{2J-1})} &= (I - R_j^T \Phi_j R_j) \mathbf{u}^{(n-1+\frac{j-1}{2J-1})} + R_j^T \Phi_j \mathbf{u}_j^{(n-\frac{1}{2})} \\ &= \sum_{s \neq j} R_s^T \Phi_s R_s \left(\sum_{l=1}^{j-1} R_l^T \Phi_l \mathbf{u}_l^{(n-\frac{1}{2})} + \sum_{l=j}^J R_l^T \Phi_l \mathbf{u}_l^{(n-1)} \right) + R_j^T \Phi_j \mathbf{u}_j^{(n-\frac{1}{2})} \\ &= \sum_{l=1}^j R_l^T \Phi_l \mathbf{u}_l^{(n-\frac{1}{2})} + \sum_{l=j+1}^J R_l^T \Phi_l \mathbf{u}_l^{(n-1)}. \end{aligned}$$

By induction, we know this holds for all $j = 1, \dots, J-1$. Similarly, the backward sweep gives

$$\mathbf{u}^{(n-1+\frac{2J-j}{2J-1})} = \sum_{l=1}^{j-1} R_l^T \Phi_l \mathbf{u}_l^{(n-\frac{1}{2})} + \sum_{l=j}^J R_l^T \Phi_l \mathbf{u}_l^{(n)} \quad \forall j = J, \dots, 1.$$

In particular, we have $\mathbf{u}^{(n)} = \sum_{l=1}^J R_l^T \Phi_l \mathbf{u}_l^{(n)}$, and the result follows by induction. \square

Remark 24. The DOSM can be used also as a preconditioner for the original problem. The DOSM preconditioner as a linear operator acts on a given r.h.s. and outputs an approximate solution by one or multiple steps of the DOSM iteration with *zero* initial guess. As we showed above, choosing an appropriate gluing scheme is very important for the preconditioned Richardson iteration $\mathbf{u}^{(l)} = \mathbf{u}^{(l-1)} + M^{-1}(\mathbf{f} - A\mathbf{u}^{(l-1)})$ to be equivalent to the DOSM in the subdomain transmission form. If we use the preconditioner with a Krylov method, e.g., GMRES, the convergence cannot be slower than the Richardson iteration. Hence, the equivalence tells us that if the optimal DOSM from Theorem 18 is used as a preconditioner for a Krylov method, the latter also converges in one step. This is not the case for the gluing scheme of the AS, which uses for the weights all ones on each subdomain. Another advantage of using the DOSM as a *preconditioner* is that even if one uses inexact subdomain solves, i.e., $\tilde{M}^{-1} \approx M^{-1}$, the global consistency is retained, i.e., the converged iterate is always the solution of the original problem, while the *plain* DOSM iterates have a consistency error.

Remark 25. From the subdomain transmission form (Algorithm 1 or 2) to the deferred correction form (Algorithm 3 or Algorithm 4), we see that the interface conditions become homogeneous and the evaluation of the r.h.s. becomes unrelated to the transmission operators \mathcal{P} , \mathcal{Q} or P , Q . This can be an advantage when the action of those operators is expensive. For example, the PML technique (see Remark 13) leads to the choice $\mathcal{Q} = \mathcal{I}$, $\mathcal{P} = \text{DtN}_j^{pml}$ on $\Gamma_{j,j\pm 1}$. In this case, the action of \mathcal{P} involves solving a problem in Ω_j^{pml} which one might want to avoid. In the deferred correction form, the action of \mathcal{P} is not required for the evaluation of the r.h.s., but \mathcal{P} still appears as acting on the unknown function in the interface condition

$$\mathbf{n}_j^T \alpha \nabla v_j + \text{DtN}_j^{pml} v_j = 0 \quad \text{on } \Gamma_{j,j\pm 1},$$

where we have omitted the superscripts for simplicity. For the implementation, one usually unfolds the PML-DtN operator and composes a bigger problem defined in $\Omega_j \cup \Omega_j^{pml}$, as explained in Remark 13. The first use of PML in a Schwarz method is due to Toselli (see [163]), who seemed to use the full extension including the PML regions for updating the global iterates, so that his algorithm deviates from the optimized Schwarz method and may be interpreted as an overlapping Schwarz method with the

PML equations used in the overlap. This resembles in the overlap the shifted-Laplace based Schwarz method recently proposed and studied in [90].

Remark 26. There are many other ways of implementing the PML transmission conditions. One was proposed in [148, 149]. First, we rewrite the condition as

$$\mathbf{n}_j^T \alpha \nabla u_j + \text{DtN}_j^{pml}(u_j - u_{j-1}) = \mathbf{n}_j^T \alpha \nabla u_{j-1} \quad \text{on } \Gamma_{j,j-1}.$$

Then, we unfold DtN_j^{pml} and compose a coupled problem in $\Omega_j \cup \Omega_j^{pml}$ as follows:

$$\begin{aligned} \mathcal{L}u_j &= f && \text{in } \Omega_j, \\ \mathcal{B}u_j &= g && \text{on } \partial\Omega_j \cap \partial\Omega, \\ \mathbf{n}_j^T \alpha \nabla u_j - \mathbf{n}_j^T \alpha \nabla v &= \mathbf{n}_j^T \alpha \nabla u_{j-1} && \text{on } \partial\Omega_j - \partial\Omega, \\ \tilde{\mathcal{L}}v &= 0 && \text{in } \Omega_j^{pml}, \\ \tilde{\mathcal{B}}v &= 0 && \text{on } \partial\Omega_j^{pml} - \partial\Omega_j, \\ v - u_j &= -u_{j-1} && \text{on } \partial\Omega_j^{pml} - \partial\Omega. \end{aligned}$$

A straightforward discretization requires two sets of d.o.f. on the interfaces $\Gamma_{j,j\pm 1}$, one for u_j and the other for v . In this way, we need only to solve one coupled problem in $\Omega_j \cup \Omega_j^{pml}$ and we avoid solving an extra problem in Ω_j^{pml} . We can further remove the Dirichlet jump from the coupled problem by extending (lifting) u_{j-1} from the interfaces into Ω_j^{pml} . Let the extended function be $\tilde{u}_{j-1}^{j,pml}$. We change the unknown in the PML to $\tilde{v} := v + \tilde{u}_{j-1}^{j,pml}$ so that the coupled problem for (u_j, \tilde{v}) enforces no jump of Dirichlet traces across the interfaces.

6.3. Substructured Form of Schwarz Methods. The substructured form of the optimized Schwarz method consists in taking interface data as iterates (unknowns).¹⁶ These iterates are substantially smaller than the volume iterates and thus can save memory and flops for Krylov subspace methods. This form was first introduced in [132] for one-way domain decompositions like in Figure 2. In particular, for Helmholtz problems, it was used in [30, 76]. Later, the substructured form was generalized to the case of domain decompositions with cross points (points shared by three or more subdomains); see, e.g., [12, 62, 73, 81, 124] for various approaches (some are called FETI-2LM). Here, we consider only the sequential one-way domain decomposition from Figure 2. The substructured form of the DOSM is given in Algorithm 5 at the PDE level and in Algorithm 6 at the matrix level. Theorems 27 and 28 give the equivalence of the substructured formulations to the formulations with subdomain iterates. Their proofs are simple and we thus omit them here.

THEOREM 27. *Algorithm 5 is equivalent to Algorithm 1. That is, given $\lambda_j^{(0)} = \mathcal{Q}_j^\rangle(\mathbf{n}_j^T \alpha \nabla u_j^{(0)}) + \mathcal{P}_j^\rangle u_j^{(0)}$, the iterates generated by the two algorithms satisfy $\lambda_j^{(n)} = \mathcal{Q}_j^\rangle(\mathbf{n}_j^T \alpha \nabla u_j^{(n)}) + \mathcal{P}_j^\rangle u_j^{(n)}$.*

THEOREM 28. *Algorithm 6 is equivalent to Algorithm 2. That is, given $\lambda_j^{(0)} = (Q_j^\rangle A_j^\rangle + P_j^\rangle) \mathbf{u}_j^{(0)} + Q_j^\rangle A_{j,j\bullet} \mathbf{u}_{j\bullet}^{(0)} - Q_j^\rangle \mathbf{f}_j^\rangle$, the iterates generated by the two algorithms satisfy $\lambda_j^{(n)} = (Q_j^\rangle A_j^\rangle + P_j^\rangle) \mathbf{u}_j^{(n)} + Q_j^\rangle A_{j,j\bullet} \mathbf{u}_{j\bullet}^{(n)} - Q_j^\rangle \mathbf{f}_j^\rangle$.*

¹⁶If exact subdomain solvers are used, the glued global approximation has compact residual for the original problem near interfaces. Making use of this property leads to yet another substructured form; see Algorithm 12 in subsection 8.2.

Algorithm 5. DOSM in the **substructured** form at the **PDE** level.

Given the last iterate $\lambda_j^{(n-1)} = \{\lambda_j^{(n-1)} \text{ on } \Gamma_{j,j+1}, j = 1, \dots, J-1\}$, solve successively for $j = 1, \dots, J-1$,

$$\begin{aligned} \mathcal{L} u_j^{(n-\frac{1}{2})} &= f && \text{in } \Omega_j, \\ \mathcal{B} u_j^{(n-\frac{1}{2})} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathcal{Q}_j^{\langle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^{\langle} u_j^{(n-\frac{1}{2})} &= \lambda_{j\langle}^{(n)} && \text{on } \Gamma_{j,j-1}, \\ \mathcal{Q}_j^{\rangle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n-\frac{1}{2})} \right) + \mathcal{P}_j^{\rangle} u_j^{(n-\frac{1}{2})} &= \lambda_{j\rangle}^{(n-1)} && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

with each solve followed by $\lambda_{j+1\langle}^{(n)} \leftarrow \mathcal{Q}_{j+1}^{\langle} \left(\mathbf{n}_{j+1}^T \alpha \nabla u_j^{(n-\frac{1}{2})} \right) + \mathcal{P}_{j+1}^{\langle} u_j^{(n-\frac{1}{2})}$. Then solve successively for $j = J, \dots, 1$,

$$\begin{aligned} \mathcal{L} u_j^{(n)} &= f && \text{in } \Omega_j, \\ \mathcal{B} u_j^{(n)} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathcal{Q}_j^{\langle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n)} \right) + \mathcal{P}_j^{\langle} u_j^{(n)} &= \lambda_{j\langle}^{(n)} && \text{on } \Gamma_{j,j-1}, \\ \mathcal{Q}_j^{\rangle} \left(\mathbf{n}_j^T \alpha \nabla u_j^{(n)} \right) + \mathcal{P}_j^{\rangle} u_j^{(n)} &= \lambda_{j\rangle}^{(n)} && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

with each solve followed by $\lambda_{j-1\rangle}^{(n)} \leftarrow \mathcal{Q}_{j-1}^{\rangle} \left(\mathbf{n}_{j-1}^T \alpha \nabla u_j^{(n)} \right) + \mathcal{P}_{j-1}^{\rangle} u_j^{(n)}$.

We obtain $\lambda_j^{(n)} \leftarrow \{\lambda_j^{(n)} \text{ on } \Gamma_{j,j+1}, j = 1, \dots, J-1\}$.

To explain how to use Krylov acceleration for the substructured formulations, we denote $\lambda_{\langle}^{(n)} := \{\lambda_{j\langle}^{(n)}, j = 2, \dots, J\}$. The forward and backward sweeps of Algorithm 5 define a linear forward map \mathcal{F}_{\langle} and a backward map \mathcal{F}_{\rangle} such that $\lambda_{\langle}^{(n)} = \mathcal{F}_{\langle}(\lambda_{\rangle}^{(n-1)}, f, g)$ and $\lambda_{\rangle}^{(n)} = \mathcal{F}_{\rangle}(\lambda_{\langle}^{(n)}, f, g)$. The corresponding fixed point equation $\lambda_{\rangle} = \mathcal{F}_{\rangle}(\mathcal{F}_{\langle}(\lambda_{\rangle}, f, g), f, g)$ can be rewritten as a linear system

$$(\mathcal{I} - \mathcal{F}_{\rangle}(\mathcal{F}_{\langle}(\cdot, 0, 0), 0, 0)) \lambda_{\rangle} = \mathcal{F}_{\rangle}(\mathcal{F}_{\langle}(0, f, g), f, g),$$

which can now be solved by polynomial methods, e.g., Chebyshev iterations and/or Krylov methods.

Remark 29. If we look at each solve and the following update, we have a linear map $\mathcal{F}_{j+1\langle}$ such that $\lambda_{j+1\langle}^{(n)} = \mathcal{F}_{j+1\langle}(\lambda_{j\langle}^{(n)}, \lambda_j^{(n-1)}, f, g)$ and $\mathcal{F}_{j-1\rangle}$ such that $\lambda_{j-1\rangle}^{(n)} = \mathcal{F}_{j-1\rangle}(\lambda_{j\langle}^{(n)}, \lambda_j^{(n)}, f, g)$. Considering the converged solution (i.e., removing the superscripts), we find the linear system

$$\left[\begin{array}{cccc|cccc} 1 & & & & -b_1^- & & & \\ -a_1^- & 1 & & & & -b_2^- & & \\ & & \ddots & & & & \ddots & \\ & & & -a_{j-2}^- & 1 & & & -b_{j-1}^- \\ -b_1^+ & & & & 1 & -a_1^+ & & \\ & -b_2^+ & & & & 1 & \ddots & \\ & & \ddots & & & & \ddots & -a_{j-2}^+ \\ & & & -b_{j-1}^+ & & & 1 & \end{array} \right] \begin{bmatrix} \lambda_{2\langle} \\ \lambda_{3\langle} \\ \vdots \\ \lambda_{j\langle} \\ \lambda_{1\rangle} \\ \lambda_{2\rangle} \\ \vdots \\ \lambda_{j-1\rangle} \end{bmatrix} = \begin{bmatrix} g_{2\langle} \\ g_{3\langle} \\ \vdots \\ g_{j\langle} \\ g_{1\rangle} \\ g_{2\rangle} \\ \vdots \\ g_{j-1\rangle} \end{bmatrix},$$

Algorithm 6. DOSM in the **substructured** form at the **matrix** level.

Given the last iterate $\lambda_j^{(n-1)} = \{\lambda_j^{(n-1)}, j = 1, \dots, J-1\}$, solve successively for $j = 1, \dots, J-1$,

$$\begin{bmatrix} Q_{j\langle}^\langle A_{j\langle}^\langle + P_{j\langle}^\langle & Q_{j\langle}^\langle A_{j\langle\bullet} & & \\ & A_{j\bullet j\langle} & A_{j\bullet\bullet} & \\ & & & A_{j\bullet j} \rangle \\ & Q_{j\langle}^\rangle A_{j\langle\bullet} & Q_{j\langle}^\rangle A_{j\langle}^\rangle + P_{j\langle}^\rangle & \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle}^{(n-\frac{1}{2})} \\ \mathbf{u}_{j\bullet}^{(n-\frac{1}{2})} \\ \mathbf{u}_{j\langle}^{(n-\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} Q_{j\langle}^\langle \mathbf{f}_{j\langle}^\langle + \lambda_{j\langle}^{(n)} \\ \mathbf{f}_{j\bullet} \\ Q_{j\langle}^\rangle \mathbf{f}_{j\langle}^\rangle + \lambda_{j\langle}^{(n-1)} \end{bmatrix},$$

with each solve followed by

$$\lambda_{j+1\langle}^{(n)} \leftarrow Q_{j+1\langle}^\langle \left(\mathbf{f}_{j+1\langle}^\langle - A_{j+1\langle j\mathbb{H}} \mathbf{u}_{j\mathbb{H}}^{(n-\frac{1}{2})} \right) + \left(P_{j+1\langle}^\langle - Q_{j+1\langle}^\langle A_{j+1\langle}^\rangle \right) \mathbf{u}_{j\langle}^{(n-\frac{1}{2})}.$$

Note that it does not matter which splits $\mathbf{f}_{j\langle} = \mathbf{f}_{j\langle}^\langle + \mathbf{f}_{j\langle}^\rangle$ and $\mathbf{f}_{j\bullet} = \mathbf{f}_{j\bullet}^\langle + \mathbf{f}_{j\bullet}^\rangle$ are used, the only difference will be the definition of the interface data. For example, one can use the simple splits $\mathbf{f}_{j\langle}^\rangle = 0$ and $\mathbf{f}_{j\bullet}^\rangle = 0$.

Then solve successively for $j = J, \dots, 1$,

$$\begin{bmatrix} Q_{j\langle}^\langle A_{j\langle}^\langle + P_{j\langle}^\langle & Q_{j\langle}^\langle A_{j\langle\bullet} & & \\ & A_{j\bullet j\langle} & A_{j\bullet\bullet} & \\ & & & A_{j\bullet j} \rangle \\ & Q_{j\langle}^\rangle A_{j\langle\bullet} & Q_{j\langle}^\rangle A_{j\langle}^\rangle + P_{j\langle}^\rangle & \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle}^{(n)} \\ \mathbf{u}_{j\bullet}^{(n)} \\ \mathbf{u}_{j\langle}^{(n)} \end{bmatrix} = \begin{bmatrix} Q_{j\langle}^\langle \mathbf{f}_{j\langle}^\langle + \lambda_{j\langle}^{(n)} \\ \mathbf{f}_{j\bullet} \\ Q_{j\langle}^\rangle \mathbf{f}_{j\langle}^\rangle + \lambda_{j\langle}^{(n)} \end{bmatrix},$$

with each solve followed by

$$\lambda_{j-1\langle}^{(n)} \leftarrow Q_{j-1\langle}^\rangle \left(\mathbf{f}_{j-1\langle}^\langle - A_{j-1\langle j\mathbb{H}} \mathbf{u}_{j\mathbb{H}}^{(n)} \right) + \left(P_{j-1\langle}^\rangle - Q_{j-1\langle}^\rangle A_{j-1\langle}^\langle \right) \mathbf{u}_{j\langle}^{(n)}.$$

We obtain $\lambda_j^{(n)} = \{\lambda_j^{(n)}, j = 1, \dots, J-1\}$.

where $\mathbf{1}$'s are the identity operators, $\mathbf{a}_j^- := \mathcal{F}_{j+2\langle}(\cdot, 0, 0, 0)$, $\mathbf{b}_j^- := \mathcal{F}_{j+1\langle}(0, \cdot, 0, 0)$, $\mathbf{a}_j^+ := \mathcal{F}_{j\langle}(0, \cdot, 0, 0)$, $\mathbf{b}_j^+ := \mathcal{F}_{j\langle}(\cdot, 0, 0, 0)$, and $\mathbf{g}_* = \mathcal{F}_*(0, 0, f, g)$. If we regard the above system as a 2-by-2 block system, the block Gauss–Seidel method (see Remark 39 for block Jacobi) leads to Algorithm 5 with each block solved exactly by forward or backward substitution. The operators \mathbf{a}_j and \mathbf{b}_j can also be represented using Green's functions based on (37), which we will see in more detail in section 8.

7. AILU and Sweeping Preconditioners. We now explain the AILU and sweeping preconditioners. To do so, we denote in (19) $\mathbf{u}_1 := \mathbf{u}_{1\mathbb{H}}$, $\mathbf{f}_1 := \mathbf{f}_{1\mathbb{H}}$, $D_1 := A_{1\mathbb{H}}$, $L_1 := [A_{1\mathbb{H}1\mathbb{H}}; 0]$, $U_1 := [A_{1\mathbb{H}1\mathbb{H}}, 0]$, and, for $j \geq 2$, $\mathbf{u}_j := [\mathbf{u}_{j-1\mathbb{H}}; \mathbf{u}_{j\mathbb{H}}]$, $\mathbf{f}_j := [\mathbf{f}_{j-1\mathbb{H}}; \mathbf{f}_{j\mathbb{H}}]$,

$$D_j := \begin{bmatrix} A_{j-1\mathbb{H}} & A_{j-1\mathbb{H}j\mathbb{H}} \\ A_{j\mathbb{H}j-1\mathbb{H}} & A_{j\mathbb{H}} \end{bmatrix}, \quad L_j := \begin{bmatrix} 0 & A_{j\mathbb{H}j\mathbb{H}} \\ 0 & 0 \end{bmatrix}, \quad U_j := \begin{bmatrix} 0 & 0 \\ A_{j\mathbb{H}j\mathbb{H}} & 0 \end{bmatrix}.$$

With this notation, (19) becomes (2). Then, based on the factorization given in (3), we can solve (5) and (6) by forward and backward substitution, which leads to Algorithm 7.

THEOREM 30. *If T_j , $j = 1, \dots, J$, are invertible, then Algorithm 7 is equivalent to Algorithm 2 with a nonoverlapping decomposition, zero initial guess, and $Q_{j\langle}^\langle = I_{j\langle}$,*

Algorithm 7. Block LU solve for the block tridiagonal system in (2).

Compute T_j 's according to (4).

Forward sweep: solve successively the subproblems

$$\begin{aligned} T_1 \mathbf{v}_1 &= \mathbf{f}_1, \\ T_j \mathbf{v}_j &= \mathbf{f}_j - L_{j-1} \mathbf{v}_{j-1}, \quad j = 2, \dots, J. \end{aligned}$$

Backward sweep: let $\mathbf{u}_J \leftarrow \mathbf{v}_J$; solve successively the subproblems

$$T_j \mathbf{u}_j = T_j \mathbf{v}_j - U_j \mathbf{u}_{j+1}, \quad j = J-1, \dots, 1.$$

$P_{j\langle}^\langle = A_{j\langle}^\langle - A_{j\langle, \sim j} A_{\sim j}^{-1} A_{\sim j, j\langle}^\langle, Q_{j\langle}^\langle = 0, P_{j\langle}^\langle = I_{j\langle}$. That is, the iterates generated by the two algorithms satisfy $\mathbf{v}_j = [\mathbf{u}_{j\langle}^{(\frac{1}{2})}; \mathbf{u}_{j\mathbb{B}}^{(\frac{1}{2})}]$ and $\mathbf{u}_j = [\mathbf{u}_{j\langle}^{(1)}; \mathbf{u}_{j\mathbb{B}}^{(1)}]$.

Proof. We first show that the T_j 's defined by (4) satisfy for $j \geq 2$,

$$(47) \quad T_j = D_j - \tilde{L}_{j-1} A_{\sim j}^{-1} \tilde{U}_{j-1},$$

where $\tilde{L}_1 := L_1$, $\tilde{U}_1 := U_1$, $\tilde{L}_{j-1} := [0, L_{j-1}]$, and $\tilde{U}_{j-1} := [0; U_{j-1}]$ for $j \geq 3$. The case of $j = 2$ follows directly by definition. Assuming (47) holds for one j , we now show that it also holds for $j+1$ instead of j . First, by Gaussian elimination we have from (47) that

$$A_{\sim j+1}^{-1} = \begin{bmatrix} A_{\sim j} & \tilde{U}_{j-1} \\ \tilde{L}_{j-1} & D_j \end{bmatrix}^{-1} = \begin{bmatrix} * & * \\ * & T_j^{-1} \end{bmatrix},$$

where $*$ represents terms not interesting to us. Therefore,

$$D_{j+1} - \tilde{L}_j A_{\sim j+1}^{-1} \tilde{U}_j = D_{j+1} - \begin{bmatrix} 0 & L_j \end{bmatrix} \begin{bmatrix} * & * \\ * & T_j^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ U_j \end{bmatrix} = T_{j+1},$$

which is (47) with j replaced by $j+1$. By induction, (47) holds for all $j \geq 2$. Note that we are considering a nonoverlapping decomposition so we can write

$$\tilde{L}_{j-1} = \begin{bmatrix} A_{j\langle, \sim j} \\ 0 \end{bmatrix}, \quad \tilde{U}_{j-1} = \begin{bmatrix} A_{\sim j, j\langle} & 0 \end{bmatrix}, \quad D_j = \begin{bmatrix} A_{j\langle} & A_{j\langle, j\bullet} \\ A_{j\bullet, j\langle} & A_{j\bullet} \end{bmatrix}.$$

Substituting the above equations into (47), we obtain for $j \geq 2$,

$$(48) \quad T_j = \begin{bmatrix} A_{j\langle} - A_{j\langle, \sim j} A_{\sim j}^{-1} A_{\sim j, j\langle} & A_{j\langle, j\bullet} \\ A_{j\bullet, j\langle} & A_{j\bullet} \end{bmatrix}.$$

Let the initial guess of Algorithm 2 be $\mathbf{u}_j^{(0)} = 0$, $j = 1, \dots, J$. Substituting this and the specified matrices P and Q into Algorithm 2, we find

$$(49) \quad \begin{bmatrix} A_{1\bullet} & A_{1\bullet, 1\langle} \\ 0 & I_{1\langle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1\bullet}^{(\frac{1}{2})} \\ \mathbf{u}_{1\langle}^{(\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{1\bullet} \\ 0 \end{bmatrix}.$$

By definition, we know $T_1 = D_1 = A_{1\mathbb{B}} = A_{1\bullet}$ and $\mathbf{f}_1 = \mathbf{f}_{1\bullet}$. Hence, from (49), we have $T_1 \mathbf{u}_{1\bullet}^{(\frac{1}{2})} = \mathbf{f}_{1\bullet}$, which is satisfied also by \mathbf{v}_1 of Algorithm 7. Since T_1 is invertible and

from the fact that the decomposition is nonoverlapping, we have $\mathbf{v}_1 = \mathbf{u}_{1\bullet}^{(\frac{1}{2})} = \mathbf{u}_{1\mathbb{H}}^{(\frac{1}{2})}$. From (49) and again using the fact that the decomposition is nonoverlapping, we have $\mathbf{u}_{1\langle}^{(\frac{1}{2})} = \mathbf{u}_{1\mathbb{H}}^{(\frac{1}{2})} = 0$. Now assume that¹⁷

$$(50) \quad \mathbf{v}_{j-1} = \left[\mathbf{u}_{j-1\langle}^{(\frac{1}{2})}; \mathbf{u}_{j-1\mathbb{H}}^{(\frac{1}{2})} \right], \quad \mathbf{u}_{j-1\langle}^{(\frac{1}{2})} = 0,$$

with \mathbf{v}_{j-1} obtained by Algorithm 7 and $\mathbf{u}_{j-1\mathbb{H}}^{(\frac{1}{2})}$ obtained by Algorithm 2. In the next substep of Algorithm 2, we substitute the specified matrices P and Q , and the second equation of (50), to find

$$(51) \quad \left[\begin{array}{cc|c} A_{j\langle} - A_{j\langle, \sim j} A_{\sim j}^{-1} A_{\sim j, j\langle} & A_{j\langle \bullet} & 0 \\ \hline A_{j\bullet} & A_{j\bullet} & A_{j\bullet j} \\ \hline 0 & 0 & I_j \end{array} \right] \left[\begin{array}{c} \mathbf{u}_{j\langle}^{(\frac{1}{2})} \\ \mathbf{u}_{j\bullet}^{(\frac{1}{2})} \\ \mathbf{u}_{j\mathbb{H}}^{(\frac{1}{2})} \end{array} \right] = \left[\begin{array}{c} \mathbf{f}_{j\langle} - A_{j\langle j-1\mathbb{H}} \mathbf{u}_{j-1\mathbb{H}}^{(\frac{1}{2})} \\ \mathbf{f}_{j\bullet} \\ 0 \end{array} \right].$$

By (48), we know that the upper-left 2-by-2 block matrix in (51) equals T_j . From the first equation of (50), we see that the first two rows of the r.h.s. of (51) equal $\mathbf{f}_2 - L_1 \mathbf{v}_1$. Given that T_j is invertible and that the decomposition is nonoverlapping, we have from (51) and Algorithm 7 that

$$(52) \quad \mathbf{v}_j = \left[\mathbf{u}_{j\langle}^{(\frac{1}{2})}; \mathbf{u}_{j\mathbb{H}}^{(\frac{1}{2})} \right], \quad \mathbf{u}_{j\langle}^{(\frac{1}{2})} = 0.$$

Thus, by induction, (52) holds for all $j = 1, \dots, J-1$ and

$$\mathbf{u}_J = \mathbf{v}_J = \left[\mathbf{u}_{J\langle}^{(1)}; \mathbf{u}_{J\mathbb{H}}^{(1)} \right].$$

In Algorithm 7 we substitute $T_j \mathbf{v}_j$ from the forward sweep into the backward sweep and obtain the equivalent backward solve

$$T_j \mathbf{u}_j = \mathbf{f}_j - L_{j-1} \mathbf{v}_{j-1} - U_j \mathbf{u}_{j+1}.$$

First, we note that the coefficient matrix T_j is the same as in the forward solve. Second, compared to the forward solve, the present r.h.s. has an extra term $-U_j \mathbf{u}_{j+1}$ which corresponds to using the new Dirichlet data taken from the neighboring subdomain on the right. So Algorithm 7 and the specified case of Algorithm 2 remain equivalent in their backward solves, and we have for $j = J, \dots, 1$ ¹⁸

$$\mathbf{u}_j = \left[\mathbf{u}_{j\langle}^{(1)}; \mathbf{u}_{j\mathbb{H}}^{(1)} \right].$$

Thus, the equivalence of the two algorithms is proved. \square

Based on Theorem 30, the PDE analogue of Algorithm 7 can be stated as Algorithm 1 with a nonoverlapping decomposition, zero initial guess, and $\mathcal{Q}_{j\langle}^{\langle} = \mathcal{I}$, $\mathcal{P}_{j\langle}^{\langle} = \text{DtN}_j^{\langle}$, $\mathcal{Q}_{j\mathbb{H}}^{\langle} = 0$, $\mathcal{P}_{j\mathbb{H}}^{\langle} = \mathcal{I}$. Next, we revisit the AILU and the sweeping preconditioners. Based on Theorem 30, it is straightforward to get the following corollaries.

¹⁷When $j = 2$, we need to remove the nonexistent block.

¹⁸When $j = 1$, we need to remove the nonexistent block.

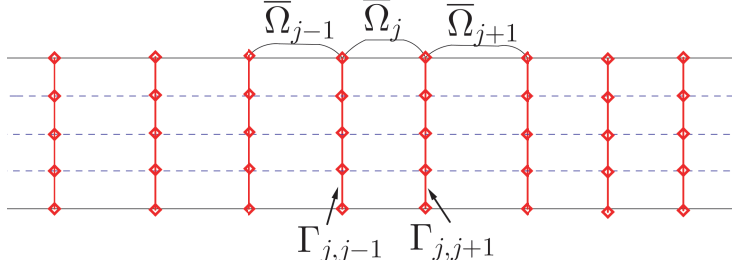


Fig. 3 Nonoverlapping domain decomposition with empty interior, $\diamond \in \Gamma_{*,\#}$.

COROLLARY 31. *The AILU preconditioner in [78] is equivalent to one step of the DOSM with a nonoverlapping decomposition such that the subdomain interiors are empty (i.e., $\Theta_j = \emptyset$; see Figure 3), zero initial guess, and second-order absorbing transmission conditions on the left interfaces and Dirichlet transmission conditions on the right interfaces of subdomains.*

COROLLARY 32. *The sweeping preconditioners in [51, 52] are equivalent to one step of the DOSM with a nonoverlapping decomposition such that the subdomain interiors are empty (i.e., $\Theta_j = \emptyset$; see Figure 3), zero initial guess, and PML or \mathcal{H} -matrix transmission conditions on the left interfaces and Dirichlet transmission conditions on the right interfaces of subdomains.*

Here we finish our discussions of the algorithms. First we point out an analogy between the LU factorization of the matrix from the discretization of $\partial_{xx} + \partial_{yy} + k^2$ and the formal analytic factorization of the differential operator,

$$(53) \quad \partial_{xx} + \partial_{yy} + k^2 = (\partial_x - i\sqrt{\partial_{yy} + k^2})(\partial_x + i\sqrt{\partial_{yy} + k^2}).$$

This analogy was first drawn in [77]. The factorization in (53) represents a reformulation of the two-way wave (Helmholtz) equation as two one-way wave equations. Hence, the original *boundary value problem* (BVP) given by the Helmholtz equation (11) can be solved in terms of certain *initial value problems* (IVPs) of the one-way wave equations. The forward substitution of the lower triangular system and the backward solution of the upper triangular system at the matrix level are related to solving these IVPs. Such analytic factorizations have been studied by many researchers; see, e.g., [100, 125]. Next, we give a derivation of the factorization based on a family of DtN operators and explain its relation to Algorithm 1. We will do so for the more general case with curved domains and curved interfaces.

We assume that the domain Ω can be described by the curvilinear coordinates,

$$\Omega = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{x} = (x_l)_{l=1}^d, x_l = x_l(\xi_1, \dots, \xi_d), \xi_1 \in (a, b) \subset \mathbb{R}, (\xi_2, \dots, \xi_d) \in Y \subset \mathbb{R}^{d-1}\},$$

with $(\frac{\partial x_j}{\partial \xi_l})$ nonsingular and x_j sufficiently smooth in $\boldsymbol{\xi}$. We may view Ω as a topological cylinder with the axial variable ξ_1 . For $s \in [a, b]$, we denote

$$\begin{aligned} \Omega^s &:= \{\mathbf{x} \in \Omega \mid x_l = x_l(\xi_1, \dots, \xi_d), \xi_1 \in (a, s), (\xi_2, \dots, \xi_d) \in Y\}, \\ \Gamma^s &:= \{\mathbf{x} \in \Omega \mid x_l = x_l(s, \xi_2, \dots, \xi_d), (\xi_2, \dots, \xi_d) \in Y\}. \end{aligned}$$

Let $d = 2$ for simplicity. In the curvilinear coordinates, the operator \mathcal{L} of (11) becomes

$$\mathcal{L}v = -\left(\frac{\partial\xi_1}{\partial x_1}\frac{\partial}{\partial\xi_1} + \frac{\partial\xi_2}{\partial x_1}\frac{\partial}{\partial\xi_2}\right)\left(\beta_1^1\frac{\partial v}{\partial\xi_1} + \beta_2^1\frac{\partial v}{\partial\xi_2}\right) - \left(\frac{\partial\xi_1}{\partial x_2}\frac{\partial}{\partial\xi_1} + \frac{\partial\xi_2}{\partial x_2}\frac{\partial}{\partial\xi_2}\right)\left(\beta_1^2\frac{\partial v}{\partial\xi_1} + \beta_2^2\frac{\partial v}{\partial\xi_2}\right) - \frac{\omega^2}{\kappa}v,$$

where $\beta_l^j = \sum_{m=1}^d \alpha_m^j \frac{\partial\xi_l}{\partial x_m}$ and (α_l^j) is the coefficient matrix α in Cartesian coordinates. We assume that \mathcal{L} can be rewritten in the form

$$\mathcal{L}v = \gamma_3 \left\{ \frac{\partial}{\partial\xi_1} \left(\gamma_1 \frac{\partial v}{\partial\xi_1} + \gamma_2 \frac{\partial v}{\partial\xi_2} + \gamma_0 v \right) - \mathcal{L}_2 v \right\},$$

with $\gamma_j = \gamma_j(\xi_1, \xi_2)$, $j = 1, 2, 3$, γ_3 and γ_1 nowhere zero, and \mathcal{L}_2 a partial differential operator w.r.t. ξ_2 only. This assumption can be verified if β_1^1, β_1^2 are sufficiently smooth or if $\xi_1 = x_1, \xi_2 = x_2$ and α_1^2 is sufficiently smooth. We define the operator-valued function $\mathcal{P}(s)$ for $s \in [a, b]$ as the s -dependent generalized DtN operator (we assume the problem below is well-posed):

$$\begin{aligned} \mathcal{P}(s) : d \rightarrow \left(\gamma_1 \frac{\partial v}{\partial\xi_1} + \gamma_2 \frac{\partial v}{\partial\xi_2} + \gamma_0 v \right) |_{\Gamma^s}, \quad \text{s.t.} \quad & \mathcal{L}v = 0 \quad \text{in } \Omega^s, \\ & \mathcal{B}v = 0 \quad \text{on } \partial\Omega^s \cap \partial\Omega, \\ & v = d \quad \text{on } \Gamma^s. \end{aligned}$$

Let us consider how $\mathcal{P}(s)$ changes with s . Let v be the solution in the definition of $\mathcal{P}(s)$. We have for $\Delta s < 0$,

$$\begin{aligned} \gamma_1 \frac{\partial v}{\partial\xi_1}(s, \xi_2) + \gamma_2 \frac{\partial v}{\partial\xi_2}(s, \xi_2) + \gamma_0 v(s, \xi_2) &= \mathcal{P}(s)v(s, \xi_2), \quad \xi_2 \in Y, \\ \gamma_1 \frac{\partial v}{\partial\xi_1}(s + \Delta s, \xi_2) + \gamma_2 \frac{\partial v}{\partial\xi_2}(s + \Delta s, \xi_2) + \gamma_0 v(s + \Delta s, \xi_2) &= \mathcal{P}(s + \Delta s)v(s + \Delta s, \xi_2), \quad \xi_2 \in Y. \end{aligned}$$

Subtracting the two equations and dividing by Δs , we get

$$\begin{aligned} & 1/\Delta s \cdot \{(\gamma_1 \partial_1 v + \gamma_2 \partial_2 v + \gamma_0 v)(s + \Delta s, \xi_2) - (\gamma_1 \partial_1 v + \gamma_2 \partial_2 v + \gamma_0 v)(s, \xi_2)\} \\ &= 1/\Delta s \cdot \{\mathcal{P}(s + \Delta s) - \mathcal{P}(s)\}v(s + \Delta s, \xi_2) + \mathcal{P}(s)\{v(s + \Delta s, \xi_2) - v(s, \xi_2)\}/\Delta s. \end{aligned}$$

We assume that the difference quotients above converge as $\Delta s \rightarrow 0$, and we find

$$(54) \quad \frac{\partial}{\partial\xi_1} \left(\gamma_1 \frac{\partial v}{\partial\xi_1} + \gamma_2 \frac{\partial v}{\partial\xi_2} + \gamma_0 v \right) (s, \xi_2) = \mathcal{P}'(s)v(s, \xi_2) + \mathcal{P}(s) \frac{\partial v}{\partial\xi_1}(s, \xi_2).$$

Using $\mathcal{L}v = 0$ and $\gamma_3 \neq 0$, we find that $\mathcal{L}_2(s, \xi_2)v(s, \xi_2)$ is equal to the l.h.s. of (54) and thus to the r.h.s. too. We further replace $\frac{\partial v}{\partial\xi_1}(s, \xi_2)$ with the definition of $\mathcal{P}(s)$ to obtain

$$\mathcal{L}_2(s, \cdot)v(s, \cdot) = \mathcal{P}'(s)v(s, \cdot) + \mathcal{P}(s) \{ \gamma_1^{-1}(s, \cdot) (\mathcal{P}(s) - \gamma_2(s, \cdot)\partial_2 - \gamma_0(s, \cdot)) v(s, \cdot) \}.$$

Removing $v(s, \cdot)$ from the above equation, we obtain the differential Riccati equation

$$(55) \quad \mathcal{P}'(s) = \mathcal{P}(s) \{ \gamma_1^{-1}(s, \cdot) (-\mathcal{P}(s) + \gamma_2(s, \cdot)\partial_2 + \gamma_0(s, \cdot)) \} + \mathcal{L}_2(s, \cdot).$$

As mentioned in [102], (55) can also be obtained from (4) when the width of the subdomain is equal to the mesh size and goes to zero. The initial value $\mathcal{P}(a)$ for (55) can be obtained from the boundary condition $\mathcal{B}v = 0$ on Γ^a if the condition is not of Dirichlet type (we assume this in the following derivation; otherwise, we should not

use the DtN operator but the NtD operator for \mathcal{P}). In particular, if $a = -\infty$ and there exists $a' \in \mathbb{R}$ such that the problem in the definition of $\mathcal{P}(s)$ is independent of $s \leq a'$, then by letting $\mathcal{P}'(s) = 0$ we obtain from the differential Riccati equation the algebraic Riccati equation

$$\mathcal{P}(s) \{ \gamma_1^{-1}(s, \cdot) (-\mathcal{P}(s) + \gamma_2(s, \cdot) \partial_2 + \gamma_0(s, \cdot)) \} + \mathcal{L}_2(s, \cdot) = 0 \quad \forall s \leq a'.$$

The solution of the algebraic Riccati equation at $s = a'$ gives us an initial value $\mathcal{P}(a')$ for the differential Riccati equation. In what follows, we assume \mathcal{P} has been precomputed on $[a, b]$.

We introduce $w := (\gamma_1 \frac{\partial u}{\partial \xi_1} + \gamma_2 \frac{\partial u}{\partial \xi_2} + \gamma_0 u) - \mathcal{P}u$ with u the solution of (11). Again, by the definition of \mathcal{P} , the initial value $w(a, \cdot)$ can be acquired from $\mathcal{B}u = g$ on Γ^a . For example, if $\mathcal{B} = \tilde{\gamma}_1 \frac{\partial}{\partial \xi_1} + \tilde{\gamma}_2 \frac{\partial}{\partial \xi_2} + \tilde{\gamma}_0$ and v is from the definition of $\mathcal{P}(a)$, i.e., $v(a, \cdot) = u(a, \cdot)$, we have $\mathcal{B}(u - v) = \tilde{\gamma}_1 \frac{\partial(u-v)}{\partial \xi_1} = g$ on Γ^a and

$$w(a, \cdot) = \gamma_1 \frac{\partial(u-v)}{\partial \xi_1}(a, \cdot) + \gamma_2 \frac{\partial(u-v)}{\partial \xi_2}(a, \cdot) + \gamma_0(u-v)(a, \cdot) = \gamma_1 \frac{\partial(u-v)}{\partial \xi_1}(a, \cdot) = \gamma_1 \tilde{\gamma}_1^{-1} g.$$

We calculate the partial derivative of w ,

$$\begin{aligned} \frac{\partial w}{\partial \xi_1} &= \frac{\partial}{\partial \xi_1} \left(\gamma_1 \frac{\partial u}{\partial \xi_1} + \gamma_2 \frac{\partial u}{\partial \xi_2} + \gamma_0 u \right) - \mathcal{P}'u - \mathcal{P} \frac{\partial u}{\partial \xi_1} \\ &= \gamma_3^{-1} f + \mathcal{L}_2 u - \mathcal{P}'u - \mathcal{P} \frac{\partial u}{\partial \xi_1} \\ &= \gamma_3^{-1} f + \mathcal{P} \{ \gamma_1^{-1} (\mathcal{P}u - \gamma_2 \partial_2 u - \gamma_0 u) \} - \mathcal{P} \frac{\partial u}{\partial \xi_1} \\ (56) \quad &= \gamma_3^{-1} f - \mathcal{P} \{ \gamma_1^{-1} w \}, \end{aligned}$$

where we have successively used the PDE satisfied by u , the differential Riccati equation of \mathcal{P} , and the definition of w . Thus, we have obtained an IVP for w which is the analogue of the “L” system of the LU factorization. The analytic analogue of the “U” system from which to recover u is simply the definition of w ,

$$(57) \quad \gamma_1 \frac{\partial u}{\partial \xi_1} = w - \gamma_2 \frac{\partial u}{\partial \xi_2} - \gamma_0 u + \mathcal{P}u.$$

The initial value for this system is set on Γ^b and can be sought again from the boundary condition $\mathcal{B}u = g$ and the one-way wave equation itself at $\xi_1 = b$. We thus have as the analytic analogue of the LU factorization

$$\frac{\partial}{\partial \xi_1} \left(\gamma_1 \frac{\partial}{\partial \xi_1} + \gamma_2 \frac{\partial}{\partial \xi_2} + \gamma_0 \right) - \mathcal{L}_2 = \left(\frac{\partial}{\partial \xi_1} + \mathcal{P} \{ \gamma_1^{-1} \cdot \} \right) \left(\gamma_1 \frac{\partial}{\partial \xi_1} + \gamma_2 \frac{\partial}{\partial \xi_2} + \gamma_0 - \mathcal{P} \right),$$

which can be verified for an arbitrary function $v(\xi_1, \xi_2)$ as follows:

$$\begin{aligned} & \left(\frac{\partial}{\partial \xi_1} + \mathcal{P} \{ \gamma_1^{-1} \cdot \} \right) \left(\gamma_1 \frac{\partial}{\partial \xi_1} + \gamma_2 \frac{\partial}{\partial \xi_2} + \gamma_0 - \mathcal{P} \right) v \\ &= \left(\frac{\partial}{\partial \xi_1} + \mathcal{P} \{ \gamma_1^{-1} \cdot \} \right) \left(\gamma_1 \frac{\partial v}{\partial \xi_1} + \gamma_2 \frac{\partial v}{\partial \xi_2} + \gamma_0 v - \mathcal{P}v \right) \\ &= \frac{\partial}{\partial \xi_1} \left(\gamma_1 \frac{\partial v}{\partial \xi_1} + \gamma_2 \frac{\partial v}{\partial \xi_2} + \gamma_0 v \right) - \frac{\partial}{\partial \xi_1} (\mathcal{P}v) + \mathcal{P} \frac{\partial v}{\partial \xi_1} + \mathcal{P} \{ \gamma_1^{-1} (\gamma_2 \frac{\partial v}{\partial \xi_2} + \gamma_0 v - \mathcal{P}v) \} \\ &= \frac{\partial}{\partial \xi_1} \left(\gamma_1 \frac{\partial v}{\partial \xi_1} + \gamma_2 \frac{\partial v}{\partial \xi_2} + \gamma_0 v \right) - \mathcal{L}_2 v, \end{aligned}$$

where to obtain the last identity we substituted the differential Riccati equation (55). Note that all of the derivation above needs to be justified in appropriate function spaces, for which we refer to, e.g., [102].

Solving the IVP in (56) is not the only way to get w . We can also solve the original problem (11) restricted to Ω^{s_1} ($s_1 \leq b$) complemented with an arbitrary boundary condition on Γ^{s_1} that guarantees well-posedness, which results in \tilde{u} . From the proof of Theorem 17, we have $w = (\gamma_1 \frac{\partial \tilde{u}}{\partial \xi_1} + \gamma_2 \frac{\partial \tilde{u}}{\partial \xi_2} + \gamma_0 \tilde{u}) - \mathcal{P}\tilde{u}$ on Γ^t for all $t \in (a, s_1]$. Supposing w is known in Ω^{s_1} , to obtain w in $\Omega^{s_2} - \Omega^{s_1}$ for $s_2 > s_1$, we only have to solve the original problem restricted to $\Omega^{s_2} - \Omega^{s_1}$, where \tilde{u} satisfies the boundary condition $(\gamma_1 \frac{\partial \tilde{u}}{\partial \xi_1} + \gamma_2 \frac{\partial \tilde{u}}{\partial \xi_2} + \gamma_0 \tilde{u}) - \mathcal{P}\tilde{u} = w$ on Γ^{s_1} and an arbitrary boundary condition for well-posedness on Γ^{s_2} ; then, we have $w = (\gamma_1 \frac{\partial \tilde{u}}{\partial \xi_1} + \gamma_2 \frac{\partial \tilde{u}}{\partial \xi_2} + \gamma_0 \tilde{u}) - \mathcal{P}\tilde{u}$ on Γ^t for all $t \in (a, s_2]$. This process continues forward until Γ^b whereupon w is obtained in Ω . Then we solve (11) restricted to $\Omega^b - \Omega^{s_{J-1}}$ for u with w providing interface data on $\Gamma^{s_{J-1}}$. To find u further backward in $\Omega^{s_{J-1}} - \Omega^{s_{J-2}}$, we again use w as interface data on $\Gamma^{s_{J-2}}$, while an arbitrary boundary condition on $\Gamma^{s_{J-1}}$, as long as the resulting problem is well-posed, can be extracted from already-known u in $\Omega^b - \Omega^{s_{J-1}}$. This process continues backward until Γ^a . The forward plus backward processes constitute exactly Algorithm 1. In other words, we may view Algorithm 1 as a way of solving the IVPs for w and u in (56) and (57).

8. Methods Motivated by Physics. We now present several algorithms motivated by various intuitions from physics and developed using Green's function techniques. We start with the special case considered in section 5, where the source term vanished outside a subdomain Ω_j , in which we showed how to truncate the original problem to Ω_j to avoid discretizing the big domain Ω . To be able to use this as a building block for a more general solver, we need two further ingredients: first, since we are now not only interested in the near-field solution u in Ω_j , but also the far-field u in $\Omega - \Omega_j$, we need to be able to map the near-field waves to the far-field. This is a classical engineering problem; see, e.g., [162, pp. 329–352]. Second, we may also have sources outside Ω_j which stimulate waves that come into Ω_j . The question is then how to incorporate the influence of these exterior sources on the local solution on the subdomain Ω_j .

In the rest of this section we assume that $g = 0$ in (11) to simplify our presentation. From the solution formula in (38), namely,

$$u(\mathbf{x}) = \int_{\Omega} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y},$$

we see that if we restrict to $\mathbf{x}, \mathbf{y} \in \Omega_j$, then the corresponding diagonal part of $G(\mathbf{x}, \mathbf{y})$ can be well approximated by a good truncation of the corresponding BVP to Ω_j . The second point raised in the last paragraph actually asks how to approximate the off-diagonal parts of $G(\mathbf{x}, \mathbf{y})$ when \mathbf{x}, \mathbf{y} are in *different* subdomains. This is a core problem studied in \mathcal{H} -matrix and similar techniques (see the references mentioned in section 4), but a direct approximation of the off-diagonal parts of the Green's function is difficult for waves traveling long-distance in heterogeneous media. As an alternative, one can first solve for the diagonal parts, i.e., the near-field waves stimulated by the sources within each subdomain, and then propagate the waves gradually subdomain by subdomain from near to far. This is possible because of the so-called equivalence theorem in engineering, which states that if we enclose a source by a surface, referred to as a Huygens surface, then the stimulated waves in the exterior can be determined

from the waves on the Huygens surface, thought of as new equivalent sources that are called currents in engineering. For example, once we have $u_{1,1} := u_{,1}|_{\Omega_1}$ with $u_{,1}(\mathbf{x}) := \int_{\Omega_1} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}$, $\mathbf{x} \in \Omega$, we should also be able to find $u_{2,1} := u_{,1}|_{\Omega_2}$. In fact, $u_{,1}$ solves the BVP

$$\mathcal{L}u_{,1} = f_1 \text{ in } \Omega, \quad \mathcal{B}u_{,1} = 0 \text{ on } \partial\Omega,$$

where $f_1 := \mathcal{E}_1(f|_{\Omega_1})$; so we deduce that $u_{2,1}$ can be obtained from

$$(58) \quad \begin{aligned} \mathcal{L}u_{2,1} &= 0 && \text{in } \Omega_2, \\ \mathcal{B}u_{2,1} &= 0 && \text{on } \partial\Omega \cap \partial\Omega_2, \\ \mathcal{B}_2^{\langle} u_{2,1} &= \mathcal{B}_2^{\langle} u_{1,1} && \text{on } \Gamma_{2,1}, \\ \mathbf{n}_2^T \alpha \nabla u_{2,1} + \text{DtN}_2 u_{2,1} &= 0 && \text{on } \Gamma_{2,3}, \end{aligned}$$

where \mathcal{B}_2^{\langle} is an arbitrary boundary operator so that the problem is well-posed, and we assume DtN_2 is well-defined. We see that the influence of f_1 on the waves in Ω_2 has been transformed to an equivalent surface current $\mathcal{B}_2^{\langle} u_{1,1}$. In summary, the near-field waves $u_{1,1}$ generate a surface current $\mathcal{B}_2^{\langle} u_{1,1}$ from which one can recover the far-field waves $u_{2,1}$, as the equivalence theorem says.

Since in Ω_2 we also want $u_{2,2}(\mathbf{x}) = \int_{\Omega_2} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}$, $\mathbf{x} \in \Omega_2$, it is convenient to add the source f_2 directly in (58) and to solve for $(u_{2,1} + u_{2,2})(\mathbf{x}) = u_{2,1:2}(\mathbf{x}) := \int_{\Omega_1 \cup \Omega_2} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}$, $\mathbf{x} \in \Omega_2$, at once from

$$(59) \quad \begin{aligned} \mathcal{L}u_{2,1:2} &= f_2 && \text{in } \Omega_2, \\ \mathcal{B}u_{2,1:2} &= 0 && \text{on } \partial\Omega \cap \partial\Omega_2, \\ \mathbf{n}_2^T \alpha \nabla u_{2,1:2} + \text{DtN}_2 u_{2,1:2} &= \mathbf{n}_2^T \alpha \nabla u_{1,1} + \text{DtN}_2 u_{1,1} && \text{on } \Gamma_{2,1}, \\ \mathbf{n}_2^T \alpha \nabla u_{2,1:2} + \text{DtN}_2 u_{2,1:2} &= 0 && \text{on } \Gamma_{2,3}, \end{aligned}$$

where we specified \mathcal{B}_2^{\langle} of (58) as the transparent boundary operator to simulate the waves generated by f_2 without spurious reflections. Using (37), the solution of (59) can be represented as

$$u_{2,1:2}(\mathbf{x}) = \int_{\Omega_2} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} + \int_{\Gamma_{2,1}} G(\mathbf{x}, \mathbf{y}) \lambda_{2\langle}(\mathbf{y}) d\sigma(\mathbf{y}), \quad \mathbf{x} \in \Omega_2,$$

where $\lambda_{2\langle} := \mathcal{B}_2^{\langle} u_{1,1} = \mathbf{n}_2^T \alpha \nabla u_{1,1} + \text{DtN}_2 u_{1,1}$. Now that $u_{2,1:2}$ contains the influence of both f_1 and f_2 , this influence can be passed on to Ω_3 through a transmission condition on $\Gamma_{3,2}$, and using a transparent transmission condition also then permits us to include the influence of f_3 locally in Ω_3 . This process continues until we obtain $u_{J,1:J}$, which is the exact solution of (11) restricted to Ω_J , i.e., $u_{J,1:J}(\mathbf{x}) = \int_{\Omega} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}$, $\mathbf{x} \in \Omega_J$. Now that we have $u_{j,1:j}$, $j = 1, \dots, J$, and the interface data $\lambda_{j+1\langle} := \mathcal{B}_{j+1}^{\langle} u_{j,1:j} = \mathbf{n}_{j+1}^T \alpha \nabla u_{j,1:j} + \text{DtN}_{j+1} u_{j,1:j}$, $j = 1, \dots, J-1$, we want to add the waves $u_{j,j+1:J}$, stimulated by the sources on the right of Ω_j , to $u_{j,1:j}$ to get $u_{j,1:J} = u_{j,1:j} + u_{j,j+1:J}$, the solution of the original BVP in (11) restricted to Ω_j . We note that $\mathcal{B}_{j+1}^{\langle} u_{j,1:J} = \mathcal{B}_{j+1}^{\langle} u_{j,1:j} = \lambda_{j+1\langle}$ because $\mathcal{B}_{j+1}^{\langle} u_{j,j+1:J} = 0$ by Lemma 10. That is, the waves from $\Omega_{j+1:J}$ pass through $\Gamma_{j+1,j}$ transparently. For $u_{J-1,1:J}$, the other interface data $\mathcal{B}_{J-1}^{\rangle} u_{J-1,1:J}$ on $\Gamma_{J-1,J}$ is available from the known solution $u_{J,1:J}$.

Therefore, $u_{J-1,1:J}$ satisfies the BVP

$$(60) \quad \begin{aligned} \mathcal{L}v_{J-1} &= f_{J-1} && \text{in } \Omega_{J-1}, \\ \mathcal{B}v_{J-1} &= 0 && \text{on } \partial\Omega \cap \partial\Omega_{J-1}, \\ \mathbf{n}_{J-1}^T \alpha \nabla v_{J-1} + \text{DtN}_{J-1} v_{J-1} &= \lambda_{J-1} \langle && \text{on } \Gamma_{J-1,J-2}, \\ \mathcal{B}_{J-1}^\rangle v_{J-1} &= \mathcal{B}_{J-1}^\rangle u_{J,1:J} && \text{on } \Gamma_{J-1,J}, \end{aligned}$$

where all the data is known, $\mathcal{B}_{J-1}^\rangle$ is arbitrary as long as (60) is well-posed, and the first two equations are just taken from (11). In other words, (60) is set up according to what the solution u of (11) satisfies, and the unique solvability of (60) justifies that its solution can only be $v_{J-1} = u_{J-1,1:J} = u|_{\Omega_{J-1}}$. Having obtained $u_{J-1,1:J}$, the exact information can be passed backward further through $\Gamma_{J-2,J-1}$. This process continues until we obtain the exact solution in all the subdomains. We can also formulate this process using the representation formula in (37). For example, if $\lambda_{J-1} \rangle := \mathcal{B}_{J-1}^\rangle u_{J,1:J} = \mathbf{n}_{J-1}^T \alpha \nabla u_{J,1:J} + \text{DtN}_{J-1} u_{J,1:J}$, then solving (60) is equivalent to computing for $\mathbf{x} \in \Omega_{J-1}$,

$$u_{J-1,1:J}(\mathbf{x}) = \int_{\Omega_{J-1}} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} + \int_{\Gamma_{J-1,J-2}} G(\mathbf{x}, \mathbf{y}) \lambda_{J-1} \langle (\mathbf{y}) d\sigma(\mathbf{y}) + \int_{\Gamma_{J-1,J}} G(\mathbf{x}, \mathbf{y}) \lambda_{J-1} \rangle (\mathbf{y}) d\sigma(\mathbf{y}).$$

We have now presented the basic ideas digested from [28, 157, 179, 180], and have again derived Algorithm 1. To propagate the waves based on physical insight, we were forced here to use absorbing transmission conditions on $\Gamma_{j,j+1}$ in the forward sweep. In the next subsections, we will explain in detail the different paths that led to the invention of the new methods in [28, 157, 179, 180]. We will see that these new methods were derived in quite different forms, but it will become clear how closely they are related to the algorithms discussed in sections 6 and 7.

8.1. The Source Transfer Method Using Equivalent Volume Sources. The source transfer method from [28] is based on a decomposition of Ω into nonoverlapping and serially connected layers O_j , $j = 0, \dots, J$, which are combined into subdomains $\Omega_j := O_{j-1} \cup \Gamma_j \cup O_j$ with $\Gamma_j := \partial O_{j-1} \cap \partial O_j$ for $j = 1, \dots, J$. We saw this decomposition in Remark 5, and we also assume here that the resulting system is block tridiagonal, as in Remark 5. The key idea of the source transfer method is the physical intuition that it is possible to transfer the sources before O_J into O_{J-1} without changing the wave field in O_J . This is done layer by layer. First, the source in O_0 is transferred to O_1 without changing the wave field to the right of O_1 , i.e., in O_j , $j = 2, \dots, J$. In terms of the Green's function, we need to find a map Ψ_1 that transfers $\tilde{f}_{1\mathfrak{q}} := f|_{O_0}$ to the source $\Psi_1(\tilde{f}_{1\mathfrak{q}})$ defined in O_1 such that

$$\int_{O_0} G(\mathbf{x}, \mathbf{y}) \tilde{f}_{1\mathfrak{q}}(\mathbf{y}) d\mathbf{y} = \int_{O_1} G(\mathbf{x}, \mathbf{y}) \Psi_1(\tilde{f}_{1\mathfrak{q}})(\mathbf{y}) d\mathbf{y} \quad \forall \mathbf{x} \in O_l, \quad l = 2, \dots, J.$$

Then we define $\tilde{f}_{2\mathfrak{q}} := f|_{O_1} + \Psi_1(\tilde{f}_{1\mathfrak{q}})$. For $j = 1, \dots, J-2$, we try to find a map Ψ_{j+1} that transfers $\tilde{f}_{j+1\mathfrak{q}}$ to the source $\Psi_{j+1}(\tilde{f}_{j+1\mathfrak{q}})$ in O_{j+1} such that for $l = j+2, \dots, J$,

$$(61) \quad \int_{O_j} G(\mathbf{x}, \mathbf{y}) \tilde{f}_{j+1\mathfrak{q}}(\mathbf{y}) d\mathbf{y} = \int_{O_{j+1}} G(\mathbf{x}, \mathbf{y}) \Psi_{j+1}(\tilde{f}_{j+1\mathfrak{q}})(\mathbf{y}) d\mathbf{y} \quad \forall \mathbf{x} \in O_l,$$

and define $\tilde{f}_{j+2\mathfrak{q}} := f|_{O_{j+1}} + \Psi_{j+1}(\tilde{f}_{j+1\mathfrak{q}})$. Eventually, we get $\tilde{f}_{J\mathfrak{q}}$ and the wave field in O_J of the original problem in (11) is given by

$$u_{J\mathfrak{q}}(\mathbf{x}) = \int_{O_J} G(\mathbf{x}, \mathbf{y}) f_{J\mathfrak{q}}(\mathbf{y}) d\mathbf{y} + \int_{O_{J-1}} G(\mathbf{x}, \mathbf{y}) \tilde{f}_{J\mathfrak{q}}(\mathbf{y}) d\mathbf{y} \quad \forall \mathbf{x} \in O_J.$$

Once $u_{J\mathfrak{q}}$ is known, it leaves a Dirichlet trace on Γ_J which, together with the transferred source $\tilde{f}_{J-1\mathfrak{q}}$ and the source in O_{J-1} , results in the wave field in O_{J-1} ,

$$\begin{aligned} u_{J-1\mathfrak{q}}(\mathbf{x}) &= \int_{O_{J-1}} G_{J-1}(\mathbf{x}, \mathbf{y}) f_{J-1\mathfrak{q}}(\mathbf{y}) d\mathbf{y} + \int_{O_{J-2}} G_{J-1}(\mathbf{x}, \mathbf{y}) \tilde{f}_{J-1\mathfrak{q}}(\mathbf{y}) d\mathbf{y} \\ &\quad + \int_{\Gamma_J} [\mathbf{n}^T \alpha^T \nabla_{\mathbf{y}} G_{J-1}(\mathbf{x}, \mathbf{y})] u_{J\mathfrak{q}}(\mathbf{y}) d\sigma(\mathbf{y}) \quad \forall \mathbf{x} \in O_{J-1}, \end{aligned}$$

where G_{J-1} is the Green's function in Ω_{J-1} satisfying a transparent boundary condition on Γ_{J-2} and a homogeneous Dirichlet condition on Γ_J . This amounts to solving a BVP in Ω_{J-1} with Dirichlet boundary condition $u_{J-1} = u_J$ on Γ_J and the transparent boundary condition on Γ_{J-2} . Afterwards, the wave field in O_{J-1} can be again extended backward to O_{J-2} . This process continues until we have obtained the wave field in all the layers.

The authors of [28] emphasize that “*the key step in the method is the source transfer algorithm*,” which results in (61). So how can one obtain the source transfer functions Ψ_{j+1} ? While the source transfer appeared very naturally in the block LU factorization in (7), it is a bit more involved to find Ψ_{j+1} in the present setting: the authors substitute the source $\tilde{f}_{j+1\mathfrak{q}}$ in O_j with the generated wave field v_{j+1} in Ω_{j+1} by using the PDE $\mathcal{L}v_{j+1} = \tilde{f}_{j+1}$ in Ω_{j+1} (let \tilde{f}_{j+1} be the zero extension of $\tilde{f}_{j+1\mathfrak{q}}$ to Ω_{j+1}). Substituting this and $-\frac{\omega^2}{\kappa}G(\mathbf{x}, \mathbf{y}) = \nabla_{\mathbf{y}}^T(\alpha^T \nabla_{\mathbf{y}} G(\mathbf{x}, \mathbf{y}))$ at $\mathbf{x} \neq \mathbf{y}$ into the l.h.s. of (61), they obtain for $\mathbf{x} \in O_l$, $l = j+2, \dots, J$,

$$\begin{aligned} &\int_{O_j} G(\mathbf{x}, \mathbf{y}) \tilde{f}_{j+1\mathfrak{q}}(\mathbf{y}) d\mathbf{y} \\ &= \int_{O_j} [-\nabla^T(\alpha \nabla v_{j+1}(\mathbf{y}))] G(\mathbf{x}, \mathbf{y}) + [\nabla_{\mathbf{y}}^T(\alpha^T \nabla_{\mathbf{y}} G(\mathbf{x}, \mathbf{y}))] v_{j+1}(\mathbf{y}) d\mathbf{y}. \end{aligned}$$

Integrating by parts and substituting the boundary conditions (transparent on Γ_j relative to O_j for both v_{j+1} and $G(\mathbf{x}, \cdot)$ with \mathbf{x} to the right of O_{j+1}) leads to

$$\begin{aligned} &\int_{O_j} G(\mathbf{x}, \mathbf{y}) \tilde{f}_{j+1\mathfrak{q}}(\mathbf{y}) d\mathbf{y} \\ (62) \quad &= \int_{\Gamma_{j+1}} -[\mathbf{n}_j^T \alpha \nabla v_{j+1}(\mathbf{y})] G(\mathbf{x}, \mathbf{y}) + [\mathbf{n}_j^T \alpha^T \nabla_{\mathbf{y}} G(\mathbf{x}, \mathbf{y})] v_{j+1}(\mathbf{y}) d\sigma(\mathbf{y}). \end{aligned}$$

The idea for transferring the source in O_j to O_{j+1} consists in a secondary integration by parts but from Γ_{j+1} to O_{j+1} . This will involve another surface integral,

$$(63) \quad \int_{\Gamma_{j+2}} [\mathbf{n}_{j+1}^T \alpha \nabla v_{j+1}(\mathbf{y})] G(\mathbf{x}, \mathbf{y}) - [\mathbf{n}_{j+1}^T \alpha^T \nabla_{\mathbf{y}} G(\mathbf{x}, \mathbf{y})] v_{j+1}(\mathbf{y}) d\sigma(\mathbf{y}).$$

Since Γ_{j+2} is *not* a transparent boundary for $G(\mathbf{x}, \mathbf{y})$, $\mathbf{y} \in O_{j+1}$, $\mathbf{x} \in O_l$ ($l > j+1$), the above surface integral in general does not vanish. Note that, however, the information

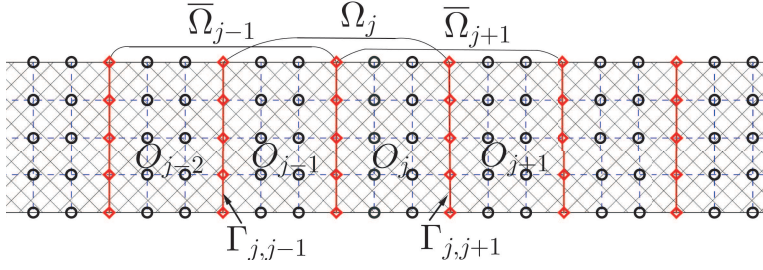


Fig. 4 Overlapping domain decomposition with contacting overlaps, $\circ \in O_*$, $\diamond \in \Gamma_{*,\#}$.

to be transferred is all from Γ_{j+1} , which is provided by the Dirichlet and Neumann traces of v_{j+1} ; see (62). So v_{j+1} can be modified in O_{j+1} without changing this information while letting the Dirichlet and Neumann traces on Γ_{j+2} vanish to get rid of (63). The authors use a function β_{j+1} for this purpose, which smoothly damps v_{j+1} from Γ_{j+1} to Γ_{j+2} such that

$$(64) \quad \begin{aligned} \beta_{j+1} &= 1, \quad (\alpha^T \mathbf{n}_j)^T \nabla \beta_{j+1} = 0 \text{ on } \Gamma_{j+1}, \quad \beta_{j+1} = 0, \quad (\alpha^T \mathbf{n}_j)^T \nabla \beta_{j+1} = 0 \text{ on } \Gamma_{j+2}, \\ \mathcal{B}(\beta_{j+1} v_{j+1}) &= 0 \text{ on } \partial O_{j+1} \cap \partial \Omega. \end{aligned}$$

Otherwise, the precise shape of the function β_{j+1} has no influence on the algorithm. Substituting $\beta_{j+1} v_{j+1}$ for v_{j+1} in the r.h.s. of (62) and (63) and summing, we find

$$\begin{aligned} & \int_{O_j} G(\mathbf{x}, \mathbf{y}) \tilde{f}_{j+1\sharp}(\mathbf{y}) \, d\mathbf{y} \\ &= \int_{\partial O_{j+1}} [\mathbf{n}^T \alpha \nabla (\beta_{j+1} v_{j+1})(\mathbf{y})] G(\mathbf{x}, \mathbf{y}) - [\mathbf{n}^T \alpha^T \nabla_{\mathbf{y}} G(\mathbf{x}, \mathbf{y})] (\beta_{j+1} v_{j+1})(\mathbf{y}) \, d\sigma(\mathbf{y}). \end{aligned}$$

Integrating by parts for the r.h.s. and using $\mathcal{M}_{\mathbf{y}} G(\mathbf{x}, \mathbf{y}) = 0$ in O_{j+1} for $\mathbf{x} \notin O_{j+1}$ yields the identity

$$\int_{O_j} G(\mathbf{x}, \mathbf{y}) \tilde{f}_{j+1\sharp}(\mathbf{y}) \, d\mathbf{y} = \int_{O_{j+1}} G(\mathbf{x}, \mathbf{y}) \left\{ \nabla^T [\alpha \nabla (\beta_{j+1} v_{j+1})(\mathbf{y})] + \frac{\omega^2}{\kappa} \beta_{j+1} v_{j+1}(\mathbf{y}) \right\} \, d\mathbf{y}.$$

The sum inside the above curly braces is thus the source transfer function $\Psi_{j+1}(\tilde{f}_{j+1\sharp})$ we were looking for; see (61). We can now define the source transfer method; see Algorithm 8 at the PDE level and Algorithm 9 at the matrix level. Despite the quite different motivating ideas, these algorithms look very similar to Algorithms 3 and 4. There is one difference though: in the forward sweep, the source transfer methods do not use the source in the right overlap O_j for the subproblem in Ω_j . One can, however, regard O_{j-1} as the physical subdomain and consider O_j to already belong to the PML region so that $O_j \cup \Omega_j^{pml}$ becomes the PML region for O_{j-1} . Then the source transfer methods can also be derived directly as DOSMs in the *local* deferred correction form, and we obtain the following equivalence results.

THEOREM 33. *Suppose the subproblems of the source transfer Algorithm 8 are well-posed. Then Algorithm 8 is equivalent to the DOSM preconditioner shown in Algorithm 1 in the specific case where each subdomain consists of two contacting overlaps (see Figure 4), and using PML transmission conditions on the interfaces in*

Algorithm 8. Source transfer preconditioner at the PDE level ([28, Algorithm 3.1–3.2]).

Input the source f and g . Let $\tilde{f}_1 \leftarrow f$ in O_0 and $\tilde{f}_1 \leftarrow 0$, otherwise. Solve successively for $j = 1, \dots, J-1$,

$$\begin{aligned} \mathcal{L} v_j &= \tilde{f}_j && \text{in } \Omega_j, \\ \mathcal{B} v_j &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathbf{n}_j^T \alpha \nabla v_j + \text{DtN}_j^{pml} v_j &= 0 && \text{on } \Gamma_{j,j-1}, \\ \mathbf{n}_j^T \alpha \nabla v_j + \text{DtN}_j^{pml} v_j &= 0 && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

where DtN_j^{pml} is defined by the PML along $\Gamma_{j,j-1}$ and $\Gamma_{j,j+1}$ (see Remark 13 for practical implementation), and after each solve we let $\tilde{f}_{j+1} \leftarrow f - \mathcal{L}(\beta_j v_j)$ in O_j , and $\tilde{f}_{j+1} \leftarrow 0$ otherwise, and β_j satisfies (64).

Let \tilde{f}_j be unchanged in O_{j-1} but $\tilde{f}_j \leftarrow f$ in O_j . Solve in order of $j = J, \dots, 1$ the problem

$$\begin{aligned} \mathcal{L} \tilde{u}_j &= \tilde{f}_j && \text{in } \Omega_j, \\ \mathcal{B} \tilde{u}_j &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathbf{n}_j^T \alpha \nabla \tilde{u}_j + \text{DtN}_j^{pml} \tilde{u}_j &= 0 && \text{on } \Gamma_{j,j-1}, \\ \tilde{u}_j &= \tilde{u}_{j+1} && \text{on } \Gamma_{j,j+1}. \end{aligned}$$

Output the global approximation $\tilde{u} \leftarrow \tilde{u}_j$ in O_j , $j = 1, \dots, J$, and $\tilde{u} \leftarrow \tilde{u}_1$ in O_0 .

Algorithm 9. Source transfer preconditioner at the matrix level.

Input the r.h.s. \mathbf{f} . Let $\tilde{\mathbf{f}}_{1\emptyset} \leftarrow \mathbf{f}_{1\emptyset}$. Solve successively for $j = 1, \dots, J-1$,

$$\begin{bmatrix} \tilde{S}_{j\langle}^{\langle} & A_{j\langle j-1\emptyset} & & & \\ A_{j-1\emptyset j\langle} & A_{j-1\emptyset} & A_{j-1\emptyset j+1\langle} & & \\ & A_{j+1\langle j-1\emptyset} & A_{j+1\langle} & A_{j+1\langle j\emptyset} & \\ & & A_{j\emptyset j+1\langle} & A_{j\emptyset} & A_{j\emptyset j\langle} \\ & & & A_{j\langle j\emptyset} & \tilde{S}_{j\langle}^{\rangle} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{j\langle} \\ \mathbf{v}_{j\emptyset} \\ \mathbf{v}_{j\lceil} \\ \mathbf{v}_{j\emptyset} \\ \mathbf{v}_{j\langle} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{f}}_{j\emptyset} \\ \mathbf{f}_{j+1\langle} \\ 0 \\ 0 \end{bmatrix},$$

where $\tilde{S}_{j\langle}^{\langle}$ and $\tilde{S}_{j\langle}^{\rangle}$ are defined by the PML (see Remark 14 for practical implementation), and after each solve we let $\tilde{\mathbf{f}}_{j+1\emptyset} \leftarrow \mathbf{f}_{j\emptyset} - A_{j\emptyset} I_{j\emptyset}^j (D_j \mathbf{v}_j) - A_{j\emptyset j+1\langle} I_{j+1\langle}^j (D_j \mathbf{v}_j)$, where D_j satisfies

$$(65) \quad I_{j\langle}^j D_j = 0, A_{j\langle j\emptyset} I_{j\emptyset}^j D_j = 0, I_{j+1\langle}^j (D_j - I_j) = 0, A_{j+1\langle j\emptyset} I_{j\emptyset}^j (D_j - I_j) = 0.$$

Solve in order of $j = J, \dots, 1$ the following problem:

$$\begin{bmatrix} \tilde{S}_{j\langle}^{\langle} & A_{j\langle j-1\emptyset} & & & \\ A_{j-1\emptyset j\langle} & A_{j-1\emptyset} & A_{j-1\emptyset j+1\langle} & & \\ & A_{j+1\langle j-1\emptyset} & A_{j+1\langle} & A_{j+1\langle j\emptyset} & \\ & & A_{j\emptyset j+1\langle} & A_{j\emptyset} & A_{j\emptyset j\langle} \\ & & & 0 & I_j \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_{j\langle} \\ \tilde{\mathbf{u}}_{j\emptyset} \\ \tilde{\mathbf{u}}_{j\lceil} \\ \tilde{\mathbf{u}}_{j\emptyset} \\ \tilde{\mathbf{u}}_{j\langle} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{f}}_{j\emptyset} \\ \mathbf{f}_{j+1\langle} \\ \mathbf{f}_{j\emptyset} \\ \tilde{\mathbf{u}}_{j+1\lceil} \end{bmatrix}.$$

Output the global approximation $\tilde{\mathbf{u}} \leftarrow \tilde{\mathbf{u}}_j$ in \bar{O}_j , $j = 1, \dots, J$, and $\tilde{\mathbf{u}} \leftarrow \tilde{\mathbf{u}}_1$ in O_0 .

the forward sweep and Dirichlet instead of PML on the right interfaces in the backward sweep; moreover, in the forward sweep the source terms in the right overlap of each subdomain are put to zero, which turns the right overlap effectively into part of the PML on the right of the subdomain.

Proof. We will prove several identities between the iterates generated by the two algorithms. Since we are considering the algorithms as preconditioners, we use a zero initial guess for Algorithm 1. We start from the iterate of Algorithm 1 for the specific case stated in the theorem:

$$\begin{aligned} \mathcal{L} u_1^{(\frac{1}{2})} &= \hat{f}_1 && \text{in } \Omega_1, \\ \mathcal{B} u_1^{(\frac{1}{2})} &= g && \text{on } \partial\Omega \cap \partial\Omega_1, \\ \mathbf{n}_1^T \alpha \nabla u_1^{(\frac{1}{2})} + \text{DtN}_1^{pml} u_1^{(\frac{1}{2})} &= 0 && \text{on } \Gamma_{1,2}, \end{aligned}$$

where $\hat{f}_j = f$ in O_{j-1} , and $\hat{f}_j = 0$ otherwise, for $j = 1, \dots, J-1$. This is exactly what we solve for v_1 in Algorithm 8. By uniqueness of the solution, we see that $u_1^{(\frac{1}{2})} = v_1$. Assuming now that $u_{j-1}^{(\frac{1}{2})} = v_{j-1}$ in O_{j-1} , we will show that this relation also holds for j replacing $j-1$. In the j th forward substep of Algorithm 1, we solve

$$(66) \quad \begin{aligned} \mathcal{L} u_j^{(\frac{1}{2})} &= \hat{f}_j && \text{in } \Omega_j, \\ \mathcal{B} u_j^{(\frac{1}{2})} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathbf{n}_j^T \alpha \nabla u_j^{(\frac{1}{2})} + \text{DtN}_j^{pml} u_j^{(\frac{1}{2})} &= \mathbf{n}_j^T \alpha \nabla u_{j-1}^{(\frac{1}{2})} + \text{DtN}_j^{pml} u_{j-1}^{(\frac{1}{2})} && \text{on } \Gamma_{j,j-1}, \\ \mathbf{n}_j^T \alpha \nabla u_j^{(\frac{1}{2})} + \text{DtN}_j^{pml} u_j^{(\frac{1}{2})} &= 0 && \text{on } \Gamma_{j,j+1}. \end{aligned}$$

We extend $\beta_{j-1} u_{j-1}^{(\frac{1}{2})}$ by zero into O_j and substitute the correction $u_j^{(\frac{1}{2})} - \beta_{j-1} u_{j-1}^{(\frac{1}{2})}$ for $u_j^{(\frac{1}{2})}$ into (66). Using (64), we find that the BVP satisfied by the correction term is

$$\begin{aligned} \mathcal{L} \left(u_j^{(\frac{1}{2})} - \beta_{j-1} u_{j-1}^{(\frac{1}{2})} \right) &= \hat{f}_j - \mathcal{L}(\beta_{j-1} u_{j-1}^{(\frac{1}{2})}) && \text{in } \Omega_j, \\ \mathcal{B} \left(u_j^{(\frac{1}{2})} - \beta_{j-1} u_{j-1}^{(\frac{1}{2})} \right) &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \left(\mathbf{n}_j^T \alpha \nabla + \text{DtN}_j^{pml} \right) \left(u_j^{(\frac{1}{2})} - \beta_{j-1} u_{j-1}^{(\frac{1}{2})} \right) &= 0 && \text{on } \Gamma_{j,j-1}, \\ \left(\mathbf{n}_j^T \alpha \nabla + \text{DtN}_j^{pml} \right) \left(u_j^{(\frac{1}{2})} - \beta_{j-1} u_{j-1}^{(\frac{1}{2})} \right) &= 0 && \text{on } \Gamma_{j,j+1}. \end{aligned}$$

Since we assumed that $u_{j-1}^{(\frac{1}{2})} = v_{j-1}$ in $O_{j-1} = \Omega_j \cap \Omega_{j-1}$, the above BVP is exactly the same as the BVP for v_j in Algorithm 8. By uniqueness of the solution, we thus deduce that $v_j = u_j^{(\frac{1}{2})} - \beta_{j-1} u_{j-1}^{(\frac{1}{2})}$ in Ω_j and, in particular, $v_j = u_j^{(\frac{1}{2})}$ in O_j . By induction, this last relation then holds for all $j = 1, \dots, J-1$. In the backward sweep of Algorithm 1, we solve the subdomain problem

$$\begin{aligned} \mathcal{L} u_j^{(1)} &= f && \text{in } \Omega_j, \\ \mathcal{B} u_j^{(1)} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathbf{n}_j^T \alpha \nabla u_j^{(1)} + \text{DtN}_j^{pml} u_j^{(1)} &= \mathbf{n}_j^T \alpha \nabla u_{j-1}^{(\frac{1}{2})} + \text{DtN}_j^{pml} u_{j-1}^{(\frac{1}{2})} && \text{on } \Gamma_{j,j-1}, \\ u_j^{(1)} &= u_{j+1}^{(1)} && \text{on } \Gamma_{j,j+1}. \end{aligned}$$

By the same argument as before, we can show that $\tilde{u}_j = u_j^{(1)}$ in O_j for $j = J, \dots, 1$ and $\tilde{u}_1 = u_1^{(1)}$ in O_0 . \square

THEOREM 34. *Assume that the subproblems of the discrete source transfer Algorithm 9 are well-posed. Then Algorithm 9 is equivalent to the discrete DOSM preconditioner shown in Algorithm 2, in the specific case at the discrete level corresponding to the case stated in Theorem 33 at the continuous level.*

Proof. In the specific case, and with zero initial guess for its use as a preconditioner, Algorithm 2 first solves

$$\begin{bmatrix} A_{1\emptyset} & A_{1\emptyset 2\langle} & & & \\ A_{2\langle 1\emptyset} & A_{2\langle} & A_{2\langle 1\emptyset} & & \\ & A_{1\emptyset 2\langle} & A_{1\emptyset} & A_{1\emptyset 1\langle} & \\ & & A_{1\langle 1\emptyset} & \tilde{S}_1^{\rangle} & \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1\emptyset}^{(\frac{1}{2})} \\ \mathbf{u}_{1\langle}^{(\frac{1}{2})} \\ \mathbf{u}_{1\emptyset}^{(\frac{1}{2})} \\ \mathbf{u}_{1\langle}^{(\frac{1}{2})} \\ \mathbf{u}_{1\langle}^{(\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{1\emptyset} \\ \mathbf{f}_{2\langle} \\ 0 \\ 0 \end{bmatrix},$$

which is exactly the same problem for \mathbf{v}_1 in Algorithm 9; so $\mathbf{u}_1^{(\frac{1}{2})} = \mathbf{v}_1$ by uniqueness of the solution. Now assuming that

$$(67) \quad \mathbf{u}_{j-1\emptyset}^{(\frac{1}{2})} = \mathbf{v}_{j-1\emptyset}, \quad \mathbf{u}_{j-1\langle}^{(\frac{1}{2})} = \mathbf{v}_{j-1\langle},$$

we will show these relations also hold for $j+1$ replacing j . In the specific version of Algorithm 2, $\mathbf{u}_j^{(\frac{1}{2})}$ solves

$$\begin{bmatrix} \tilde{S}_j^{\langle} & A_{j\langle j-1\emptyset} & & & \\ A_{j-1\emptyset j\langle} & A_{j-1\emptyset} & A_{j-1\emptyset j+1\langle} & & \\ & A_{j+1\langle j-1\emptyset} & A_{j+1\langle} & A_{j+1\langle j\emptyset} & \\ & & A_{j\emptyset j+1\langle} & A_{j\emptyset} & A_{j\emptyset j\langle} \\ & & & A_{j\langle j\emptyset} & \tilde{S}_j^{\rangle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle}^{(\frac{1}{2})} \\ \mathbf{u}_{j\emptyset}^{(\frac{1}{2})} \\ \mathbf{u}_{j\langle}^{(\frac{1}{2})} \\ \mathbf{u}_{j\emptyset}^{(\frac{1}{2})} \\ \mathbf{u}_{j\langle}^{(\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j\langle} + (\tilde{S}_j^{\langle} - A_{j\langle}) \mathbf{u}_{j-1\langle}^{(\frac{1}{2})} - A_{j\langle j-2\emptyset} \mathbf{u}_{j-1\emptyset}^{(\frac{1}{2})} \\ \mathbf{f}_{j\emptyset} \\ \mathbf{f}_{j\langle} \\ 0 \\ 0 \end{bmatrix}.$$

We can further rewrite the above system into the equivalent system

$$\begin{bmatrix} \tilde{S}_j^{\langle} & A_{j\langle j-1\emptyset} & & & \\ A_{j-1\emptyset j\langle} & A_{j-1\emptyset} & A_{j-1\emptyset j+1\langle} & & \\ & A_{j+1\langle j-1\emptyset} & A_{j+1\langle} & A_{j+1\langle j\emptyset} & \\ & & A_{j\emptyset j+1\langle} & A_{j\emptyset} & A_{j\emptyset j\langle} \\ & & & A_{j\langle j\emptyset} & \tilde{S}_j^{\rangle} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j\langle}^{(\frac{1}{2})} - I_{j\langle}^{j-1}(D_{j-1}\mathbf{u}_{j-1}^{(\frac{1}{2})}) \\ \mathbf{u}_{j\emptyset}^{(\frac{1}{2})} - I_{j-1\emptyset}^{j-1}(D_{j-1}\mathbf{u}_{j-1}^{(\frac{1}{2})}) \\ \mathbf{u}_{j\langle}^{(\frac{1}{2})} \\ \mathbf{u}_{j\emptyset}^{(\frac{1}{2})} \\ \mathbf{u}_{j\langle}^{(\frac{1}{2})} \end{bmatrix} \\ = \begin{bmatrix} 0 \\ \mathbf{f}_{j-1\emptyset} - A_{j-1\emptyset} I_{j-1\emptyset}^{j-1}(D_{j-1}\mathbf{u}_{j-1}^{(\frac{1}{2})}) - A_{j-1\emptyset j\langle} I_{j\langle}^{j-1}(D_{j-1}\mathbf{u}_{j-1}^{(\frac{1}{2})}) \\ \mathbf{f}_{j+1\langle} \\ 0 \\ 0 \end{bmatrix},$$

where for the first row we have used $I_{j\langle}^{j-1}(D_{j-1}\mathbf{u}_{j-1}^{(\frac{1}{2})}) = \mathbf{u}_{j-1\langle}^{(\frac{1}{2})}$, $A_{j\langle j-1\emptyset} I_{j-1\emptyset}^{j-1}(D_{j-1}\mathbf{u}_{j-1}^{(\frac{1}{2})}) = A_{j\langle j-1\emptyset} \mathbf{u}_{j-1\emptyset}^{(\frac{1}{2})}$ from (65), and $A_{j\langle j-2\emptyset} \mathbf{u}_{j-1\emptyset}^{(\frac{1}{2})} + A_{j\langle} \mathbf{u}_{j-1\langle}^{(\frac{1}{2})} + A_{j\langle j-1\emptyset} \mathbf{u}_{j-1\emptyset}^{(\frac{1}{2})} = \mathbf{f}_{j\langle}$ from the system for $\mathbf{u}_{j-1}^{(\frac{1}{2})}$. By (67) this is exactly the same problem for \mathbf{v}_j in Algorithm 9. By induction, we have $\mathbf{u}_{j\emptyset}^{(\frac{1}{2})} = \mathbf{v}_{j\emptyset}$ and $\mathbf{u}_{j\langle}^{(\frac{1}{2})} = \mathbf{v}_{j\langle}$ for all $j = 1, \dots, J-1$.

By similar arguments, we can also show that in the backward sweep, $\mathbf{u}_{j\emptyset}^{(1)} = \tilde{\mathbf{u}}_{j\emptyset}$ and $\mathbf{u}_{j\lceil}^{(1)} = \tilde{\mathbf{u}}_{j\lceil}$ for all $j = J, \dots, 2$, and $\mathbf{u}_1^{(1)} = \tilde{\mathbf{u}}_1$. \square

8.2. The Method Using Single Layer Potentials. Stolk summarizes in [157] the main ideas for the method based on single layer potentials as follows: “A new domain decomposition method is introduced for the heterogeneous 2-D and 3-D Helmholtz equations. Transmission conditions based on the perfectly matched layer (PML) are derived that avoid artificial reflections and match incoming and outgoing waves at the subdomain interfaces.”

To motivate the method based on single layer potentials, we consider the free space problem in a homogeneous medium. In this case, the source in each subdomain generates waves biased in two directions, forward and backward, which both are outgoing relative to the subdomain. In each subdomain, we are interested in the sum of the waves stimulated by *all* the sources, including the sources from the other subdomains. We must thus account for the waves incoming from all the other subdomains in each subdomain. The idea of the algorithm based on single layer potentials is forward propagation and accumulation of the waves from the first subdomain to the last one so that the waves in Ω_j , $1 \leq j \leq J$, have accounted for all the forward going waves generated from the sources in Ω_l for all $l < j$. Then, a similar procedure can be carried out backward so that the waves in Ω_j also contain the backward going waves generated from the sources in Ω_m for all $m > j$. The actual backward sweep in [157], however, solves for the correction from the residual.

To transfer the waves, the author in [157] states, “We have constructed new transmission conditions ... These are designed to ensure that:

- (i) the boundary conditions at the subdomain interfaces are non-reflecting;
- (ii) if Ω_{j-1} and Ω_j are neighboring subdomains, then the outgoing wave field from Ω_{j-1} equals the incoming wave field in Ω_j at the joint boundary and vice versa.

This is achieved in a simple and accurate way using a PML boundary layer added to the subdomains and single layer potentials.” As noted in [157], a related approach has been studied in [148] that is also motivated by matching the incoming and outgoing waves. There is, however, a difference in the concrete forms used to achieve the matching; see Remark 26 for a brief review of the form used in [148].

The representation of incoming waves is a well-studied topic in computational electromagnetics; see [162, pp. 185–220]. The common idea is to represent the incoming wave v_{j-1} from Ω_{j-1} to Ω_j as an equivalent source term on the surface $\Gamma_{j,j-1}$. In [157], the proposed equivalent source is $2\delta(s_{j-1})\mathbf{n}_j^T \alpha \nabla v_{j-1}$; here s_{j-1} is a local coordinate normal to $\Gamma_{j,j-1}$ and $s_{j-1} = 0$ corresponds to $\Gamma_{j,j-1}$, and $\delta(s_{j-1})$ here represents a surface delta function. The author in [157] states:¹⁹

a short intuitive explanation goes as follows. The term $v_{j-1}|_{\Gamma_{j,j-1}}$ exclusively contains forward going waves because of the presence of a PML non-reflecting layer immediately to its right.²⁰ The term $2\delta(s_{j-1})\mathbf{n}_j^T \alpha \nabla v_{j-1}$ is meant to cause the same forward going wave field in the field v_j as in the field v_{j-1} the source generates waves propagating both forwardly and backwardly in a symmetric fashion. The factor 2 is introduced so that the forward propagating part equals v_{j-1} on $\Gamma_{j,j-1}$. The backward propagating part is absorbed in the neighboring PML layer [along $\Gamma_{j,j-1}$ for Ω_j].

¹⁹We replaced the original notation in the quote by the notation used in this article.

²⁰In fact, the PML is placed on the right of $\Gamma_{j-1,j}$ for Ω_{j-1} .

Let us take a closer look at the reasoning above: we want to extend the waves v_{j-1} from Ω_{j-1} to Ω_j . By (37), we have for $\mathbf{x} \in \Omega_j$,

$$(68) \quad v_{j-1}(\mathbf{x}) = \int_{\Gamma_{j,j-1}} (\mathbf{n}_j^T \alpha \nabla v_{j-1})(\mathbf{y}) G(\mathbf{x}, \mathbf{y}) - \mathbf{n}_j^T(\mathbf{y}) (\alpha^T(\mathbf{y}) \nabla_{\mathbf{y}} G(\mathbf{x}, \mathbf{y})) v_{j-1}(\mathbf{y}) d\sigma(\mathbf{y}).$$

The Green's function $G(\mathbf{x}, \mathbf{y})$ represents the wave field at $\mathbf{y} \in \Gamma_{j,j-1}$ stimulated by a point source at $\mathbf{x} \in \Omega_j$. For the free space problem in a homogeneous medium, if two point sources at \mathbf{x} and \mathbf{x}' are symmetric w.r.t. $\Gamma_{j,j-1}$, then the stimulated waves generated by the individual point sources are also symmetric w.r.t. $\Gamma_{j,j-1}$.²¹ Hence,

$$(69) \quad \forall \mathbf{x} \in \Omega_j, \mathbf{x}' \text{ and } \mathbf{x} \text{ symmetric w.r.t. } \Gamma_{j,j-1}, \forall \mathbf{y} \in \Gamma_{j,j-1} : \\ G(\mathbf{x}, \mathbf{y}) = G(\mathbf{x}', \mathbf{y}), \mathbf{n}_j^T(\mathbf{y}) (\alpha^T(\mathbf{y}) \nabla_{\mathbf{y}} (G(\mathbf{x}, \mathbf{y}) + G(\mathbf{x}', \mathbf{y}))) = 0.$$

Since both v_{j-1} and $G(\mathbf{x}', \cdot)$ satisfy homogeneous Helmholtz equations in Ω_j and represent outgoing waves to the right of $\Gamma_{j,j+1}$, by using Green's identity we obtain

$$\int_{\Gamma_{j,j-1}} \mathbf{n}_j^T(\mathbf{y}) (\alpha^T(\mathbf{y}) \nabla_{\mathbf{y}} G(\mathbf{x}', \mathbf{y})) v_{j-1}(\mathbf{y}) d\sigma(\mathbf{y}) = \int_{\Gamma_{j,j-1}} \mathbf{n}_j^T(\mathbf{y}) (\alpha \nabla v_{j-1}(\mathbf{y})) G(\mathbf{x}', \mathbf{y}) d\sigma(\mathbf{y}).$$

Substituting this and (69) into (68), we find the single layer potential representation

$$v_{j-1}(\mathbf{x}) = \int_{\Gamma_{j,j-1}} 2 \mathbf{n}_j^T(\mathbf{y}) (\alpha \nabla v_{j-1}(\mathbf{y})) G(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}), \quad \mathbf{x} \in \Omega_j,$$

which is equivalent to being stimulated by the surface source

$$v_{j-1}(\mathbf{x}) = \int_{\Omega_j} 2\delta(s_{j-1}(\mathbf{y})) \mathbf{n}_j^T(\mathbf{y}) (\alpha \nabla v_{j-1}(\mathbf{y})) G(\mathbf{x}, \mathbf{y}) d\mathbf{y}, \quad \mathbf{x} \in \Omega_j.$$

This, under the symmetry assumption (69), justifies the equivalent source proposed in [157]. One can also calculate the representation in closed form; cf. [112].

The forward sweep in the method based on single layer potentials is performed up to the last subdomain Ω_J . Then, a global approximation is defined by setting $v := v_j$ in Ω_j , $j = 1, \dots, J$ (the subdomains are nonoverlapping), and a deferred correction problem will then be solved in the backward sweep. Note that v has in general jumps across interfaces between subdomains, and the residual $\tilde{f} := f - \mathcal{L}u$ involves a very singular distribution—the derivative of the surface delta function, $\delta'(s_{j-1})$. To avoid the potential obscurity of deciding to which subdomain such a singular distribution on an *interface* belongs, it is suggested in [157] to use for the backward sweep another set of nonoverlapping subdomains that contains these distribution residuals in the interior of the subdomains. These residuals are then taken as new sources which stimulate the correction wave field. A procedure similar to the forward sweep but from the last subdomain to the first one is carried out to find an approximation of the correction. Adding the correction to the wave field previously obtained with the forward sweep gives a global approximate solution of the original problem, which finalizes the definition of the preconditioner based on single layer potentials. Using two sets of subdomains could potentially double the cost of factorization. This situation can be avoided by the following trick from [157]: for the first set of subdomains the PML along $\Gamma_{j,j+1}$ begins with using the original operator near $\Gamma_{j,j+1}$, and only after

²¹This is like in the method of images for solving PDEs in a half space; cf. [177].

a certain distance, say, one mesh cell, does it change to the PML modified operator. The second set of subdomains is defined by moving $\Gamma_{j,j+1}$ forward by one mesh cell, but keeping the PML augmented region $\Omega_j \cup \Omega_j^{pml}$ the same as before (i.e., just moving outward the interfaces that separate Ω_j and Ω_j^{pml} so that Ω_j gets bigger and Ω_j^{pml} gets smaller). In this way, every subdomain matrix is unchanged from the forward sweep to the backward sweep and the same factorization can be used.

We summarize the preconditioner based on single layer potentials in Algorithm 10 at the PDE level and in Algorithm 11 at the matrix level. For the matrix version, we give two variants: the first one, originally presented in [157], is especially designed for the five point (or seven point in three dimensions) finite difference scheme with u discretized at integer grid points and the continuous interfaces located at half grid points. The second form is motivated by finite element methods with the discrete interfaces superposed on the continuous interfaces. While the first matrix form can be explained as a special discretization of the PDE Algorithm 10, the second matrix form is equivalent (under some assumptions) to the first matrix form, and unlike the PDE form the second matrix form uses the same partition for the forward and the backward sweeps. Both matrix forms do the extension and restriction like ASH does; see Remark 20.

Algorithm 10. Single layer potential preconditioner at the PDE level.

Input the source terms f and g . Suppose the decomposition is nonoverlapping.

Solve successively for $j = 1, \dots, J$,

$$\begin{aligned} \mathcal{L} v_j &= f + 2\delta(s_{j-1})\mathbf{n}_j^T \alpha \nabla v_{j-1} && \text{in } \Omega_j, \\ \mathcal{B} v_j &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathbf{n}_j^T \alpha \nabla v_j + \text{DtN}_j^{pml} v_j &= 0 && \text{on } \Gamma_{j,j-1}, \\ \mathbf{n}_j^T \alpha \nabla v_j + \text{DtN}_j^{pml} v_j &= 0 && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

where DtN_j^{pml} is the PML-DtN operator along $\Gamma_{j,j-1}$ and $\Gamma_{j,j+1}$ (see Remark 13 for practical implementation), s_{j-1} is a local coordinate normal to $\Gamma_{j,j-1}$ and $s_{j-1} = 0$ corresponds to $\Gamma_{j,j-1}$, and $\delta(s_{j-1})$ here represents a surface delta function. The PML for Ω_j along $\Gamma_{j,j+1}$ has a small starting zone between $\Gamma_{j,j+1}$, and $\Gamma_{j,j+1}^h$ where the original operator \mathcal{L} is used.

Let $v \leftarrow v_j$ in Ω_j , $j = 1, \dots, J$, and compute the residual $\tilde{f} \leftarrow f - \mathcal{L}v$ in Ω .

Denote by $\tilde{\Omega}_j$ the resulting subdomain by moving forward the boundaries $\Gamma_{j,j+1}$ to $\Gamma_{j,j+1}^h$ and $\Gamma_{j,j-1}$ to $\Gamma_{j,j-1}^h$. In order of $j = J-1, \dots, 1$, solve the problem

$$\begin{aligned} \mathcal{L} w_j &= \tilde{f} + 2\delta(s_j)\mathbf{n}_j^T \alpha \nabla w_{j+1} && \text{in } \tilde{\Omega}_j, \\ \mathcal{B} w_j &= g && \text{on } \partial\Omega \cap \partial\tilde{\Omega}_j, \\ \mathbf{n}_j^T \alpha \nabla w_j + \text{DtN}_j^{pml} w_j &= 0 && \text{on } \Gamma_{j,j-1}^h, \\ \mathbf{n}_j^T \alpha \nabla w_j + \text{DtN}_j^{pml} w_j &= 0 && \text{on } \Gamma_{j,j+1}^h, \end{aligned}$$

where the DtN_j^{pml} on $\Gamma_{j,j+1}^h$ ($\Gamma_{j,j-1}^h$) uses the PML as the subset (superset) of the PML for v_j on $\Gamma_{j,j+1}$ ($\Gamma_{j,j-1}$) that starts later (earlier using the original operator).

Output $\tilde{u} \leftarrow v + w_j$ in Ω_j , $j = 1, \dots, J-1$, and $\tilde{u} \leftarrow v$ in $\tilde{\Omega}_J$.

THEOREM 35. *Suppose the subproblems of Algorithm 10 are well-posed. If the PML-DtN operators on the two sides of each interface are equal, i.e., $\text{DtN}_j^{pml} = \text{DtN}_{j+1}^{pml}$ on $\Gamma_{j,j+1}$ and on $\Gamma_{j,j+1}^h$, then the single layer potential preconditioner as shown in Algorithm 10 is equivalent to one iteration of Algorithm 1 with zero initial*

Algorithm 11. Single layer potential preconditioner at the **matrix** level.

Input the r.h.s. \mathbf{f} . Suppose the Ω_j 's are nonoverlapping. Choose one of the following two forms (see Theorem 36 for their equivalence under certain conditions).

Form 1. Extend the nonoverlapping subdomains one layer beyond each interface. We indicate the extra d.o.f. beyond the left interfaces of the j th subdomain by the subscripts $j\langle\langle$ or $j-1[[$, and similarly for $j\rangle\rangle$ or $j+1]]$.

Solve successively for $j = 1, \dots, J$,

$$\begin{bmatrix} \tilde{S}_{j\langle\langle} & A_{j\langle\langle j\langle} & & & & \\ A_{j\langle j\langle\langle} & A_{j\langle} & A_{j\langle j\bullet} & & & \\ & A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\rangle} & & \\ & & A_{j\rangle j\bullet} & A_{j\rangle} & A_{j\rangle j\rangle\rangle} & \\ & & & A_{j\rangle j\rangle} & \tilde{S}_{j\rangle\rangle} & \end{bmatrix} \begin{bmatrix} \mathbf{v}_{j\langle\langle} \\ \mathbf{v}_{j\langle} \\ \mathbf{v}_{j\bullet} \\ \mathbf{v}_{j\rangle} \\ \mathbf{v}_{j\rangle\rangle} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{j-1}[[+ A_{j\langle\langle j\langle} \mathbf{v}_{j-1}[\\ \mathbf{f}_{j\langle} - A_{j\langle j\langle\langle} \mathbf{v}_{j-1}[[- \mathbf{v}_{j-1}[\\ \mathbf{f}_{j\bullet} \\ 0 \\ 0 \end{bmatrix},$$

where $\tilde{S}_{j\langle\langle}$ and $\tilde{S}_{j\rangle\rangle}$ are the Schur complements from the PML (see Remark 14 for practical implementation).

Let $\mathbf{v} \leftarrow \sum_{j=1}^J R_j^T (I_j^{\langle\langle} I_{j\langle}^j + I_j^{\bullet} I_{j\bullet}^j) \mathbf{v}_j$ and compute $\tilde{\mathbf{f}} \leftarrow \mathbf{f} - A\mathbf{v}$. Let $\mathbf{w}_J \leftarrow 0$.

In order of $j = J-1, \dots, 1$, solve the correction problem

$$\begin{bmatrix} \tilde{S}_{j\langle\langle} & A_{j\langle\langle j\langle} & & & & \\ A_{j\langle j\langle\langle} & A_{j\langle} & A_{j\langle j\bullet} & & & \\ & A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\rangle} & & \\ & & A_{j\rangle j\bullet} & A_{j\rangle} & A_{j\rangle j\rangle\rangle} & \\ & & & A_{j\rangle j\rangle} & \tilde{S}_{j\rangle\rangle} & \end{bmatrix} \begin{bmatrix} \mathbf{w}_{j\langle\langle} \\ \mathbf{w}_{j\langle} \\ \mathbf{w}_{j\bullet} \\ \mathbf{w}_{j\rangle} \\ \mathbf{w}_{j\rangle\rangle} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tilde{\mathbf{f}}_{j\bullet} \\ \tilde{\mathbf{f}}_{j\rangle} - A_{j\rangle j\rangle\rangle} \mathbf{w}_{j+1}] - \mathbf{w}_{j+1}] \\ \mathbf{w}_{j+1}] + A_{j\rangle j\rangle} \mathbf{w}_{j+1}] \end{bmatrix}.$$

Compute the output $\tilde{\mathbf{u}} \leftarrow \mathbf{v} + \sum_{j=1}^{J-1} R_j^T (I_j^{\bullet} I_{j\bullet}^j + I_j^{\langle\langle} I_{j\langle}^j) \mathbf{w}_j$.

Form 2. Solve successively for $j = 1, \dots, J$,

$$\begin{bmatrix} \tilde{S}_{j\langle} & A_{j\langle j\bullet} & \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\rangle} \\ & A_{j\rangle j\bullet} & \tilde{S}_{j\rangle} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{j\langle} \\ \mathbf{v}_{j\bullet} \\ \mathbf{v}_{j\rangle} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j\langle} - 2A_{j\langle j-1\bullet} \mathbf{v}_{j-1\bullet} - A_{j\langle} \mathbf{v}_{j-1}[\\ \mathbf{f}_{j\bullet} \\ 0 \end{bmatrix},$$

where $\tilde{S}_{j\langle}$ and $\tilde{S}_{j\rangle}$ are the Schur complements from the PML (see Remark 14 for practical implementation).

Let $\mathbf{v} \leftarrow \sum_{j=1}^J R_j^T (I_j^{\langle\langle} I_{j\langle}^j + I_j^{\bullet} I_{j\bullet}^j) \mathbf{v}_j$ and compute $\tilde{\mathbf{f}} \leftarrow \mathbf{f} - A\mathbf{v}$. Let $\mathbf{w}_J \leftarrow 0$.

In order of $j = J-1, \dots, 1$, solve the correction problem

$$\begin{bmatrix} \tilde{S}_{j\langle} & A_{j\langle j\bullet} & \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\rangle} \\ & A_{j\rangle j\bullet} & \tilde{S}_{j\rangle} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{j\langle} \\ \mathbf{w}_{j\bullet} \\ \mathbf{w}_{j\rangle} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{f}}_{j\bullet} \\ \tilde{\mathbf{f}}_{j\rangle} - 2A_{j\rangle j+1\bullet} \mathbf{w}_{j+1\bullet} - A_{j\rangle} \mathbf{w}_{j+1}] \end{bmatrix}.$$

Compute the output $\tilde{\mathbf{u}} \leftarrow \mathbf{v} + \sum_{j=1}^{J-1} R_j^T (I_j^{\bullet} I_{j\bullet}^j + I_j^{\langle\langle} I_{j\langle}^j) \mathbf{w}_j$.

guess, $\mathcal{Q}_j^{\langle} = \mathcal{I}$, $\mathcal{P}_j^{\langle} = \text{DtN}_j^{\text{pml}}|_{\Gamma_{j,j-1}^h}$, $\mathcal{Q}_j^{\rangle} = \mathcal{I}$, $\mathcal{P}_j^{\rangle} = \text{DtN}_j^{\text{pml}}|_{\Gamma_{j,j+1}^h}$, and using the two nonoverlapping partitions as in Algorithm 10, one partition for the forward sweep and the other for the backward sweep.

Proof. By the zero initial guess and the specific conditions for Algorithm 1 in the theorem, the $(j-1)$ st subproblem of Algorithm 1 imposes the following condition on

$\Gamma_{j-1,j} = \Gamma_{j,j-1}$, since the partition is nonoverlapping:

$$\mathbf{n}_{j-1}^T \alpha \nabla v_{j-1} + \text{DtN}_{j-1}^{pml} v_{j-1} = 0.$$

Note that $\mathbf{n}_j = -\mathbf{n}_{j-1}$ on $\Gamma_{j,j-1}$. Substituting these conditions and the assumption $\text{DtN}_j^{pml} = \text{DtN}_{j-1}^{pml}$ into the transmission condition on $\Gamma_{j,j-1}$ of the j th subproblem, we find

$$\mathbf{n}_j^T \alpha \nabla v_j + \text{DtN}_j^{pml} v_j = 2\mathbf{n}_j^T \alpha \nabla v_{j-1},$$

which imposes a Neumann jump between Ω_j and the PML on the other side of $\Gamma_{j,j-1}$. We recover the forward sweep of Algorithm 10 by moving the Neumann jump to the r.h.s. of the PDE as a surface source. Between the forward and the backward sweeps, Algorithm 10 takes the residual \tilde{f} and introduces overlaps of the old subdomains used in the forward sweep and the new subdomains used in the backward sweep. This gives the deferred correction form as in Algorithm 3 but mixed with the single layer potential on $\Gamma_{j,j+1}^h$ too. Then we can conclude by adapting the proof of Theorem 21 and the same arguments as in the forward sweep. \square

THEOREM 36. *Let Algorithm 4 use $Q_{j\langle}^{\langle} = I_{j\langle}$, $Q_{j\rangle}^{\rangle} = I_{j\rangle}$, $P_{j\langle}^{\langle} = \tilde{S}_{j\langle}^{\langle} - A_{j\langle}^{\langle}$, $P_{j\rangle}^{\rangle} = \tilde{S}_{j\rangle}^{\rangle} - A_{j\rangle}^{\rangle}$, and let the initial guess be zero. The following statements about Algorithm 11 then hold:*

- 1° *Suppose the subproblems of the first form of Algorithm 11 are well-posed and $\tilde{S}_{j\langle\langle}^{\langle}$, $\tilde{S}_{j\rangle\rangle}^{\rangle}$ are invertible. Let $\tilde{S}_{j\langle}^{\langle} := A_{j\langle} - A_{j\langle j\langle\langle} (\tilde{S}_{j\langle\langle}^{\langle})^{-1} A_{j\langle\langle j\langle}$, and $\tilde{S}_{j\rangle}^{\rangle} := A_{j\rangle} - A_{j\rangle j\rangle\rangle} (\tilde{S}_{j\rangle\rangle}^{\rangle})^{-1} A_{j\rangle\rangle j\rangle}$. If $I_{j\langle\langle} = I_{j\langle} = -A_{j\langle j\langle\langle}$, $\tilde{S}_{j\langle\langle}^{\langle} = \tilde{S}_{j-1\rangle}^{\rangle}$ for $j = 2, \dots, J$, $I_{j\rangle\rangle} = I_{j\rangle} = -A_{j\rangle j\rangle\rangle}$, $\tilde{S}_{j\rangle\rangle}^{\rangle} = \tilde{S}_{j+1\langle}^{\langle}$ for $j = 1, \dots, J-1$, and $A_{j\langle j-1\bullet} = [0, A_{j\langle j\langle\langle}]$, then the first form of Algorithm 11 is equivalent to the harmonic extension variant of Algorithm 4; see Remark 20.*
- 2° *Suppose the subproblems of the second form of Algorithm 11 are well-posed. If $\tilde{S}_{j\langle}^{\langle} = \tilde{S}_{j-1\rangle}^{\rangle}$ for $j = 2, \dots, J$ and $\tilde{S}_{j\rangle}^{\rangle} = \tilde{S}_{j+1\langle}^{\langle}$ for $j = 1, \dots, J-1$, then the second form of Algorithm 11 is equivalent to the harmonic extension variant of Algorithm 4; see Remark 20.*

Proof. We first prove claim 1°. We eliminate the first and the last rows of the j th forward subproblem and substitute with the assumptions of 1° to obtain

$$(70) \quad \begin{bmatrix} \tilde{S}_{j\langle}^{\langle} & A_{j\langle j\bullet} \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\rangle} \\ & A_{j\rangle j\bullet} & \tilde{S}_{j\rangle}^{\rangle} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{j\langle} \\ \mathbf{v}_{j\bullet} \\ \mathbf{v}_{j\rangle} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j\langle} - A_{j\langle j\langle\langle} (\mathbf{v}_{j-1\langle} + (\tilde{S}_{j\langle\langle}^{\langle} - A_{j\langle}) \mathbf{v}_{j-1\langle} + (\tilde{S}_{j\langle\langle}^{\langle})^{-1} \mathbf{v}_{j-1\langle} [-\mathbf{v}_{j-1\langle}]) \\ \mathbf{f}_{j\bullet} \\ 0 \end{bmatrix}.$$

It can be shown that the underlined expression on the right above vanishes: in fact, the above subproblem is also used with $j-1$ replacing j . In particular, the last row of the $(j-1)$ st subproblem reads

$$(71) \quad A_{j-1\rangle j-1\bullet} \mathbf{v}_{j-1\bullet} + \tilde{S}_{j-1\rangle}^{\rangle} \mathbf{v}_{j-1\rangle} = 0.$$

Recalling that $j-1\rangle$ and $j\langle$ correspond to the same d.o.f., because the decomposition is nonoverlapping, and the assumptions that $A_{j\langle j-1\bullet} = [0, A_{j\langle j\langle\langle}]$, $A_{j\langle j\langle\langle} = -I_{j\langle\langle}$ and $\tilde{S}_{j-1\rangle}^{\rangle} = \tilde{S}_{j\langle\langle}^{\langle}$, we obtain from (71) that $\tilde{S}_{j\langle\langle}^{\langle} \mathbf{v}_{j-1\langle} = \mathbf{v}_{j-1\langle}$. This shows that

the underlined expression in (70) is zero. Similar to (26), (70) is almost the same subproblem used in Algorithm 1 but (70) puts zero on the last row of the r.h.s., which is a trait of the harmonic extension variant. By reusing some arguments from the proof of Theorem 23, we can show the equivalence to Algorithm 4 and conclude with claim 1°.

To prove claim 2°, the key is to show that the r.h.s. from the second form of Algorithm 11 is the same as the r.h.s. from Algorithm 2 except that the former puts zeros on the right interfaces in the forward sweep, and on the left interfaces in the backward sweep. Similar to (26), we may write the first row of the r.h.s. from the forward sweep of Algorithm 2 as

$$\mathbf{f}_{j\langle} - A_{j\langle j-1\bullet} \mathbf{u}_{j-1\bullet}^{(\frac{1}{2})} + (\tilde{S}_{j\langle} - A_{j\langle}) \mathbf{u}_{j-1\lceil}^{(\frac{1}{2})}.$$

Using the assumption that $\tilde{S}_{j\langle} = \tilde{S}_{j-1\langle}^{\rangle}$ and the last row (with zeroed r.h.s.) of the $(j-1)$ st subproblem similar to (71), we see that the above expression is equal to

$$\mathbf{f}_{j\langle} - 2A_{j\langle j-1\bullet} \mathbf{u}_{j-1\bullet}^{(\frac{1}{2})} - A_{j\langle} \mathbf{u}_{j-1\lceil}^{(\frac{1}{2})}.$$

This is exactly the same as in the forward sweep of Algorithm 11. The remaining part of the proof of claim 2° can now be completed as in the proof of Theorem 23. \square

There is a final ingredient used in [157] based on the idea of right preconditioning. A preconditioner M^{-1} such as the one defined by Algorithm 11 can be used on either the left or the right of the original operator A . For right preconditioning of (17), one first uses an iterative method like Krylov or Richardson to solve $AM^{-1}\mathbf{r} = \mathbf{f}$ for \mathbf{r} , and then obtains the solution \mathbf{u} of (17) by computing $\mathbf{u} = M^{-1}\mathbf{r}$. Let $\mathbf{r}^{(n)}$ for $n \geq 0$ be the iterates for \mathbf{r} . Denote $\mathbf{v}^{(n+1)} := M^{-1}\mathbf{r}^{(n)}$. It can be shown that if $\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{u}^{(0)}$ and $\mathbf{r}^{(n)}$ and $\mathbf{u}^{(n)}$ are generated by the Richardson iterations

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} + \mathbf{f} - AM^{-1}\mathbf{r}^{(n)}, \quad \mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + M^{-1}(\mathbf{f} - A\mathbf{u}^{(n)}),$$

then we have the relation $\mathbf{u}^{(n)} = \mathbf{v}^{(n)} + (I - M^{-1}A)^{n-1}\mathbf{u}^{(0)}$ for $n \geq 1$. There is also a relation between the GMRES iterates for the left and the right preconditioned systems; see [146].

If one solves the restricted version of the original problem exactly in the interior of the subdomains using a direct solver, and then glues the resulting local approximations into a global approximation, then the global approximation mostly has a zero residual, except where the residual is influenced by values of the global approximation from different subdomains; see, e.g., [71]. This is why in Algorithm 10, the intermediate residual \tilde{f} is concentrated in the neighborhood of the interfaces $\Gamma_{j,j-1}$, $j \geq 2$, and the output \tilde{u} leaves the residual $f - \mathcal{L}\tilde{u}$ concentrated in the neighborhood of the shifted interfaces $\Gamma_{j,j-1}^h$, $j \geq 2$. In Algorithm 11, a component of $\tilde{\mathbf{f}}$ is nonzero only if the corresponding row in the matrix A has at least one nonzero entry belonging to a column associated with an interface d.o.f., and the residual left by the output $\tilde{\mathbf{u}}$ also has a similar sparsity. The sparsity of the residuals can be leveraged in the right preconditioned system because the essential unknowns become the nonzero components of the residual. This was studied in detail in [107] and was also suggested by Stolk in [157] for the preconditioner based on single layer potentials. We summarize the sparse residual algorithm in Algorithm 12 and justify it in Theorem 37. Note that this substructured form can be adapted to all the preconditioners resulting in sparse resid-

uals, e.g., Algorithm 4. Compared to Algorithm 6, the reduced system in Algorithm 12 is typically of twice the size but free of the need to apply the PML-DtN operators.

Algorithm 12. Residual substructuring when most rows of $I - AM^{-1}$ vanish.

Construct the 0-1 matrix R_r such that $(I - R_r^T R_r)(I - AM^{-1}) = 0$ and $R_r R_r^T = I_r$. Set an initial guess $\mathbf{u}^{(0)}$ such that $(I - R_r^T R_r)(\mathbf{f} - A\mathbf{u}^{(0)}) = 0$, e.g., $\mathbf{u}^{(0)} \leftarrow M^{-1}\mathbf{f}$. Let $\mathbf{h}_r \leftarrow R_r(\mathbf{f} - A\mathbf{u}^{(0)})$. Solve (approximately) the substructured system for \mathbf{r}_r :

$$(72) \quad (R_r A M^{-1} R_r^T) \mathbf{r}_r = \mathbf{h}_r.$$

Let $\mathbf{u} \leftarrow M^{-1} R_r^T \mathbf{r}_r + \mathbf{u}^{(0)}$, which is (approximately) the solution of (17).

THEOREM 37. *If A and M^{-1} are invertible, then the substructured system in (72) is well-posed, and if, in addition, (72) is solved exactly, then the output \mathbf{u} from Algorithm 12 is indeed the solution of (17).*

Proof. We first assume that (72) has at least one solution. Hence,

$$R_r^T R_r A M^{-1} R_r^T \mathbf{r}_r = R_r^T \mathbf{h}_r = R_r^T R_r (\mathbf{f} - A\mathbf{u}^{(0)}) = \mathbf{f} - A\mathbf{u}^{(0)},$$

where the last equality follows from the assumption on $\mathbf{u}^{(0)}$. By the assumptions on R_r , we also have

$$(I - R_r^T R_r) A M^{-1} R_r^T \mathbf{r}_r = (I - R_r^T R_r) R_r^T \mathbf{r}_r = 0.$$

Summing the two identities above, we obtain

$$A M^{-1} R_r^T \mathbf{r}_r = \mathbf{f} - A\mathbf{u}^{(0)},$$

or $A\mathbf{u} = A(M^{-1} R_r^T \mathbf{r}_r + \mathbf{u}^{(0)}) = \mathbf{f}$; that is, the output of Algorithm 12 is indeed the solution of (17). Now if $\mathbf{h}_r = 0$, then by the assumption on $\mathbf{u}^{(0)}$ we have $A M^{-1} R_r^T \mathbf{r}_r = \mathbf{f} - A\mathbf{u}^{(0)} = 0$, which, since A and M^{-1} are invertible, implies $R_r^T \mathbf{r}_r = 0$ and further $\mathbf{r}_r = 0$ using $R_r R_r^T = I_r$. Hence, (72) is well-posed. \square

8.3. Method of Polarized Traces Using Single and Double Layer Potentials.

We have already seen that (68) can be used to propagate the wave field in Ω_{j-1} to Ω_j . The only data we take from the previous subdomain are the Neumann and Dirichlet traces on the interface $\Gamma_{j,j-1}$. The method of polarized traces introduced in [180] iterates the Neumann and Dirichlet traces from neighboring subdomains in the substeps of the forward and backward sweeps, and upon completion of a double sweep, a global approximation is constructed using the representation formula (37) in subdomains where the volume potentials have been precomputed before the sweeps.²² We summarize the polarized traces preconditioner in Algorithm 13 at the PDE level. The relation to Algorithm 1 is shown in Theorem 38.

THEOREM 38. *Suppose the subproblems used for the v_j^0 in Algorithm 13 are well-posed. Let $\{u_j^{(\frac{1}{2})}\}_{j=1}^{J-1}$ and $\{u_j^{(1)}\}_{j=1}^J$ be generated by Algorithm 1 with zero initial guess, with \mathcal{Q} being the identity and \mathcal{P} equal to the PML-DtN operators. Let $u_j^{(\frac{1}{2})} := 0$*

²²In the full paper [179] that followed [180], a substructured system for the traces is first solved instead of the original system. For brevity, we will describe only the global form preconditioner, from which the corresponding substructured system is easy to derive.

Algorithm 13. Polarized traces preconditioner at the PDE level [180, Algorithm 1].

Input the source term f and assume for simplicity $g = 0$ in (11).

Suppose Ω is decomposed into nonoverlapping subdomains.

Independently for $j = 1, \dots, J$ solve

$$\begin{aligned} \mathcal{L} v_j^0 &= f_j && \text{in } \Omega_j, \\ \mathcal{B} v_j^0 &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathbf{n}_j^T \alpha \nabla v_j^0 + \text{DtN}_j^{\text{pml}} v_j^0 &= 0 && \text{on } \Gamma_{j,j-1}, \\ \mathbf{n}_j^T \alpha \nabla v_j^0 + \text{DtN}_j^{\text{pml}} v_j^0 &= 0 && \text{on } \Gamma_{j,j+1}, \end{aligned}$$

where $f_j := f|_{\Omega_j}$; see Remark 13 for a practical implementation of the PML-DtN operators. Denote by $G_j(\mathbf{x}, \mathbf{y})$ the Green's function for the subproblem above. We have $v_j^0(\mathbf{x}) = \int_{\Omega_j} G_j(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}$ for $\mathbf{x} \in \Omega_j$.

Let $\lambda_{1\langle}^D \leftarrow 0$, $\lambda_{1\langle}^N \leftarrow 0$. Successively for $j = 2, \dots, J$ compute for all $\mathbf{x} \in \Gamma_{j,j-1}$,

$$\begin{aligned} \lambda_{j\langle}^D(\mathbf{x}) &\leftarrow (\mathcal{S}_j \lambda_{j-1\langle}^N)(\mathbf{x}) - (\mathcal{D}_j \lambda_{j-1\langle}^D)(\mathbf{x}) + v_{j-1}^0(\mathbf{x}), \\ \lambda_{j\langle}^N(\mathbf{x}) &\leftarrow (\mathcal{D}_j^* \lambda_{j-1\langle}^N)(\mathbf{x}) - (\mathcal{N}_j \lambda_{j-1\langle}^D)(\mathbf{x}) + (\mathbf{n}_j^T \alpha \nabla v_{j-1}^0)(\mathbf{x}), \end{aligned}$$

where the four surface potentials vanish for $j = 2$ and are given for $j \geq 3$ by

$$\begin{aligned} (S_j w)(\mathbf{x}) &:= \int_{\Gamma_{j-1,j-2}} w(\mathbf{y}) G_{j-1}(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}), \\ (\mathcal{D}_j w)(\mathbf{x}) &:= \int_{\Gamma_{j-1,j-2}} w(\mathbf{y}) (\mathbf{n}_{j-1}^T \alpha^T)(\mathbf{y}) \nabla_{\mathbf{y}} G_{j-1}(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}), \\ (\mathcal{D}_j^* w)(\mathbf{x}) &:= \int_{\Gamma_{j-1,j-2}} w(\mathbf{y}) (\mathbf{n}_j^T \alpha)(\mathbf{x}) \nabla_{\mathbf{x}} G_{j-1}(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}), \\ (\mathcal{N}_j w)(\mathbf{x}) &:= \int_{\Gamma_{j-1,j-2}} w(\mathbf{y}) (\mathbf{n}_j^T \alpha)(\mathbf{x}) \nabla_{\mathbf{x}} \left\{ (\mathbf{n}_{j-1}^T \alpha^T)(\mathbf{y}) \nabla_{\mathbf{y}} G_{j-1}(\mathbf{x}, \mathbf{y}) \right\} d\sigma(\mathbf{y}). \end{aligned} \tag{73}$$

Let $\lambda_{j\langle}^D \leftarrow 0$, $\lambda_{j\langle}^N \leftarrow 0$. Successively for $j = J-1, \dots, 1$ compute for all $\mathbf{x} \in \Gamma_{j,j+1}$,

$$\begin{aligned} \lambda_{j\langle}^D(\mathbf{x}) &\leftarrow (\mathcal{S}_j \lambda_{j+1\langle}^N)(\mathbf{x}) - (\mathcal{D}_j \lambda_{j+1\langle}^D)(\mathbf{x}) + v_{j+1}^0(\mathbf{x}), \\ \lambda_{j\langle}^N(\mathbf{x}) &\leftarrow (\mathcal{D}_j^* \lambda_{j+1\langle}^N)(\mathbf{x}) - (\mathcal{N}_j \lambda_{j+1\langle}^D)(\mathbf{x}) + (\mathbf{n}_j^T \alpha \nabla v_{j+1}^0)(\mathbf{x}), \end{aligned}$$

where the four surface potentials vanish for $j = J-1$ and are given for $j \leq J-2$ by

$$\begin{aligned} (S_j w)(\mathbf{x}) &:= \int_{\Gamma_{j+1,j+2}} w(\mathbf{y}) G_{j+1}(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}), \\ (\mathcal{D}_j w)(\mathbf{x}) &:= \int_{\Gamma_{j+1,j+2}} w(\mathbf{y}) (\mathbf{n}_{j+1}^T \alpha^T)(\mathbf{y}) \nabla_{\mathbf{y}} G_{j+1}(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}), \\ (\mathcal{D}_j^* w)(\mathbf{x}) &:= \int_{\Gamma_{j+1,j+2}} w(\mathbf{y}) (\mathbf{n}_j^T \alpha)(\mathbf{x}) \nabla_{\mathbf{x}} G_{j+1}(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}), \\ (\mathcal{N}_j w)(\mathbf{x}) &:= \int_{\Gamma_{j+1,j+2}} w(\mathbf{y}) (\mathbf{n}_j^T \alpha)(\mathbf{x}) \nabla_{\mathbf{x}} \left\{ (\mathbf{n}_{j+1}^T \alpha^T)(\mathbf{y}) \nabla_{\mathbf{y}} G_{j+1}(\mathbf{x}, \mathbf{y}) \right\} d\sigma(\mathbf{y}). \end{aligned} \tag{74}$$

Recover independently the subdomain solutions for $\mathbf{x} \in \Omega_j$, $j = 1, \dots, J$, by

$$\begin{aligned} v_j(\mathbf{x}) &\leftarrow v_j^0(\mathbf{x}) + \int_{\Gamma_{j,j-1}} \lambda_{j\langle}^N(\mathbf{y}) G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}) - \int_{\Gamma_{j,j-1}} \lambda_{j\langle}^D(\mathbf{y}) (\mathbf{n}_j^T \alpha^T)(\mathbf{y}) \nabla_{\mathbf{y}} G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}) \\ &\quad + \int_{\Gamma_{j,j+1}} \lambda_{j\langle}^N(\mathbf{y}) G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}) - \int_{\Gamma_{j,j+1}} \lambda_{j\langle}^D(\mathbf{y}) (\mathbf{n}_j^T \alpha^T)(\mathbf{y}) \nabla_{\mathbf{y}} G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}). \end{aligned} \tag{75}$$

Output the global approximation $\tilde{u} \leftarrow v_j$ in Ω_j , $j = 1, \dots, J$.

be defined on Ω_J . We have for Algorithm 13 $\lambda_{j\zeta}^D = u_{j-1}^{(\frac{1}{2})}$, $\lambda_{j\zeta}^N = \mathbf{n}_j^T \alpha \nabla u_{j-1}^{(\frac{1}{2})}$ on $\Gamma_{j,j-1}$, and $\lambda_j^D = u_{j+1}^{(1)} - u_{j+1}^{(\frac{1}{2})} + v_{j+1}^0$, $\lambda_j^N = \mathbf{n}_j^T \alpha \nabla (u_{j+1}^{(1)} - u_{j+1}^{(\frac{1}{2})} + v_{j+1}^0)$ on $\Gamma_{j,j+1}$. Therefore, $v_j = u_j^{(1)}$ in Ω_j .

Proof. For simplicity, we consider only the case $g = 0$ in (11). According to Algorithms 1 and 13, $u_1^{(\frac{1}{2})} = v_1^0$ in Ω_1 and so

$$(76) \quad \lambda_{2\zeta}^D = u_1^{(\frac{1}{2})}, \quad \lambda_{2\zeta}^N = \mathbf{n}_2^T \alpha \nabla u_1^{(\frac{1}{2})} \quad \text{on } \Gamma_{2,1}.$$

From the algorithms, we also have for any fixed $\mathbf{x} \in \Omega_j$, $j = 2, \dots, J$,

$$\begin{aligned} (\mathbf{n}_j^T \alpha \nabla + \text{DtN}_j^{pml})(u_j^{(\frac{1}{2})} - u_{j-1}^{(\frac{1}{2})}) &= (\mathbf{n}_j^T \alpha^T \nabla + \text{DtN}_j^{pml*})G_j(\mathbf{x}, \cdot) = 0 \text{ on } \Gamma_{j,j-1}, \\ (\mathbf{n}_j^T \alpha \nabla + \text{DtN}_j^{pml})u_j^{(\frac{1}{2})} &= (\mathbf{n}_j^T \alpha^T \nabla + \text{DtN}_j^{pml*})G_j(\mathbf{x}, \cdot) = 0 \text{ on } \Gamma_{j,j+1}, \end{aligned}$$

where DtN_j^{pml*} is similar to DtN_j^{pml} but using α^T instead of α . By the representation formula in (37) on $\bar{\Omega}_j \cup \bar{\Omega}_{j\zeta}^{pml}$, where $\Omega_{j\zeta}^{pml}$ is the PML region along $\Gamma_{j,j+1}$, we have

$$\begin{aligned} u_j^{(\frac{1}{2})}(\mathbf{x}) &= \int_{\Omega_j} f(\mathbf{y}) G_j(\mathbf{x}, \mathbf{y}) d\mathbf{y} + \int_{\Gamma_{j,j-1}} \mathbf{n}_j^T \alpha \nabla u_j^{(\frac{1}{2})}(\mathbf{y}) G_j(\mathbf{x}, \mathbf{y}) - \mathbf{n}_j^T \alpha^T \nabla_{\mathbf{y}} G_j(\mathbf{x}, \mathbf{y}) u_j^{(\frac{1}{2})}(\mathbf{y}) d\sigma(\mathbf{y}) \\ (77) \quad &= v_j^0(\mathbf{x}) + \int_{\Gamma_{j,j-1}} (\mathbf{n}_j^T \alpha \nabla + \text{DtN}_j^{pml})u_j^{(\frac{1}{2})}(\mathbf{y}) G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}) \end{aligned}$$

$$(78) \quad = v_j^0(\mathbf{x}) + \int_{\Gamma_{j,j-1}} (\mathbf{n}_j^T \alpha \nabla + \text{DtN}_j^{pml})u_{j-1}^{(\frac{1}{2})}(\mathbf{y}) G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y})$$

$$(79) \quad = v_j^0(\mathbf{x}) + \int_{\Gamma_{j,j-1}} G_j(\mathbf{x}, \mathbf{y}) \mathbf{n}_j^T \alpha \nabla u_{j-1}^{(\frac{1}{2})}(\mathbf{y}) - u_{j-1}^{(\frac{1}{2})}(\mathbf{y}) \mathbf{n}_j^T \alpha^T \nabla_{\mathbf{y}} G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}),$$

where no integrals show up on the other boundaries of $\bar{\Omega}_j \cup \bar{\Omega}_{j\zeta}^{pml}$ because the boundary conditions there are homogeneous. Equation (77) is obtained by substituting the PML condition for $G_j(\mathbf{x}, \cdot)$ and using the following identity for any fixed $\mathbf{x} \in \Omega_j$ and any trace $v(\mathbf{y})$ (which can be proved by the definition of DtN_j^{pml}):

$$(80) \quad \int_{\Gamma_{j,j-1}} v(\mathbf{y}) \text{DtN}_j^{pml*} G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}) = \int_{\Gamma_{j,j-1}} G_j(\mathbf{x}, \mathbf{y}) \text{DtN}_j^{pml} v(\mathbf{y}) d\sigma(\mathbf{y}).$$

Equation (78) follows from the transmission conditions, and (79) is obtained by applying (80) and again substituting the PML condition for $G_j(\mathbf{x}, \cdot)$. Assuming that

$$(81) \quad \lambda_{j\zeta}^D = u_{j-1}^{(\frac{1}{2})}, \quad \lambda_{j\zeta}^N = \mathbf{n}_j^T \alpha \nabla u_{j-1}^{(\frac{1}{2})} \quad \text{on } \Gamma_{j,j-1},$$

we substitute them into (79) and taking Dirichlet and Neumann traces of $u_j^{(\frac{1}{2})}$ on $\mathbf{x} \in \Gamma_{j+1,j}$, we find (81) holds for $j+1$ replacing j . By induction based on (76), we conclude that (81) holds for all $j = 2, \dots, J$. Inserting (81) into (77) yields (for $j = J$ change the l.h.s. to $u_J^{(1)}$)

$$(82) \quad u_j^{(\frac{1}{2})}(\mathbf{x}) = v_j^0(\mathbf{x}) + \int_{\Gamma_{j,j-1}} G_j(\mathbf{x}, \mathbf{y}) \lambda_{j\zeta}^N(\mathbf{y}) - \lambda_{j\zeta}^D(\mathbf{y}) \mathbf{n}_j^T \alpha^T \nabla_{\mathbf{y}} G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}).$$

In particular, we have $u_J^{(1)} = v_J$ with v_J from Algorithm 13.

In the backward sweep of Algorithm 1, we denote $w_j := u_j^{(1)} - u_j^{(\frac{1}{2})} + v_j^0$, $j = J-1, \dots, 1$, and $w_J := u_J^{(1)}$. We find w_j satisfies the PML conditions, homogeneous on $\Gamma_{j,j-1}$ but inhomogeneous on $\Gamma_{j,j+1}$. By arguments similar to the last paragraph, we can show for all $j = J-1, \dots, 1$ that

$$\lambda_j^D = w_{j+1}, \quad \lambda_j^N = \mathbf{n}_j^T \alpha \nabla w_{j+1} \quad \text{on } \Gamma_{j,j+1},$$

and further

$$(83) \quad w_j(\mathbf{x}) = v_j^0(\mathbf{x}) + \int_{\Gamma_{j,j+1}} G_j(\mathbf{x}, \mathbf{y}) \lambda_j^N(\mathbf{y}) - \lambda_j^D(\mathbf{y}) \mathbf{n}_j^T \alpha^T \nabla_{\mathbf{y}} G_j(\mathbf{x}, \mathbf{y}) d\sigma(\mathbf{y}).$$

Combining (82) and (83), we conclude that $u_j^{(1)} = v_j$ with v_j from Algorithm 13. \square

Remark 39. Algorithm 13 includes a new technique not present in Algorithm 1. First, note that the local solutions of the original problem can be represented as sums of the left-going and right-going waves. Furthermore, the two parts can be simulated independently of each other; that is, the backward sweep of Algorithm 13 can be performed in parallel with the forward sweep, whereas the backward sweep of Algorithm 1 aims to simulate the total waves and thus needs to wait for the forward sweep to finish. Algorithm 1 can be modified in the same spirit: just use the data from the original problem for an initial solve on the subdomains, zero the left interface data in the backward sweep, add the approximations from the forward/backward sweep, and subtract that from the initial solve to find the total waves. Alternatively, for the block 2-by-2 interface system in Remark 29 use block Jacobi instead of Gauss–Seidel. Similar techniques were proposed in [139, 158]. For Algorithm 13, the waves from the forward/backward sweep have different *polarized* directions and they are propagated through their Dirichlet and Neumann *traces*, which gives the name of the method.

Remark 40. In the preceding sections, we did not discuss in detail solvers for the subproblems. Typically, LU factorizations are precomputed before the iteration starts, and they are then reused for the different r.h.s. in the iterative procedure. Even for the substructured forms, the typical way is *not* to precompute the interface operators explicitly (see Remark 29), but only to implement them as matrix actions through the subdomain LU solves. The reason is two-fold: first, to build an interface operator in a naive way, we need to solve the subdomain problem as many times as the number of d.o.f. on the interface; second, the resulting matrix is dense and a naive multiplication with a vector is not cheap. However, developments of low rank formats of matrices such as \mathcal{H} -matrices have greatly improved the situation. For example, for the Laplace equation, the method in [88] reduces the building cost to $\mathcal{O}(N^{2-(2/d)})$ and the application cost to $\mathcal{O}(N^{1/2})$ in two dimensions and $\mathcal{O}(N)$ in three dimensions, and some tests for the Helmholtz equation are also performed. In [179, 180], low rank techniques are used for building and applying the surface potentials in (73)–(75) for Algorithm 13.

To bring Algorithm 13 to the matrix level, we first translate the representation formula from (37) into matrix language. Suppose G is the matrix analogue of the Green’s function, i.e.,

$$(84) \quad \begin{bmatrix} G_e & G_{eb} & G_{ei} \\ G_{be} & G_b & G_{bi} \\ G_{ie} & G_{ib} & G_i \end{bmatrix} \begin{bmatrix} A_e & A_{eb} & A_{bi} \\ A_{be} & A_b & A_{bi} \\ A_{ib} & A_i & \end{bmatrix} = \begin{bmatrix} I_e & & \\ & I_b & \\ & & I_i \end{bmatrix},$$

where the rows and columns with the subscripts containing e may all be empty. Let \mathbf{u} satisfy

$$(85) \quad \begin{bmatrix} \tilde{A}_b & A_{bi} \\ A_{ib} & A_i \end{bmatrix} \begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b + \boldsymbol{\lambda}_b \\ \mathbf{f}_i \end{bmatrix}.$$

Then we have the following representation formula for \mathbf{u}_i .

PROPOSITION 41. *If (84) holds, then (85) implies*

$$(86) \quad \mathbf{u}_i = G_i \mathbf{f}_i + G_{ib} \mathbf{f}_b + G_{ib} \boldsymbol{\lambda}_b^N - \{G_{ib} A_b^{(i)} + G_i A_{ib}\} \mathbf{u}_b,$$

where $\boldsymbol{\lambda}_b^N := A_b^{(i)} \mathbf{u}_b + A_{bi} \mathbf{u}_i - \mathbf{f}_b$ and $A_b^{(i)}$ is an arbitrary matrix of appropriate size.

Proof. Inserting the definition of $\boldsymbol{\lambda}_b^N$ and the last row of (85) into the r.h.s. of (86) we obtain

$$\text{r.h.s. of (86)} = (G_i A_i + G_{ib} A_{bi}) \mathbf{u}_i.$$

On the other hand, from Equation (84) we have $G_i A_i + G_{ib} A_{bi} = I_i$, which turns the above equation into (86). \square

From (86), we can recognize $[G_i, G_{ib}]$, G_{ib} , and $G_{ib} A_b^{(i)} + G_i A_{ib}$ as the volume, single layer, and double layer potentials. We now give the matrix analogue²³ of Algorithm 13 in Algorithm 14 and prove its equivalence to Algorithm 2 in Theorem 42.

THEOREM 42. *Suppose the subproblems for the \mathbf{v}_j^0 in Algorithm 14 are well-posed. Let $\{\mathbf{u}_j^{(\frac{1}{2})}\}_{j=1}^{J-1}$ and $\{\mathbf{u}_j^{(1)}\}_{j=1}^J$ be generated by Algorithm 2 with zero initial guess, with Q equal to the identity and with P equal to the PML-DtN operators. Let $\mathbf{u}_j^{(\frac{1}{2})} := 0$. We have for Algorithm 14 $\boldsymbol{\lambda}_{j\langle}^D = \mathbf{u}_{j-1\langle}^{(\frac{1}{2})}$, $\boldsymbol{\lambda}_{j\langle}^N = -A_{j\langle} \mathbf{u}_{j-1\langle}^{(\frac{1}{2})} - A_{j\langle j-1\bullet} \mathbf{u}_{j-1\bullet}^{(\frac{1}{2})}$, and $\boldsymbol{\lambda}_j^D = \mathbf{u}_{j+1\langle}^{(1)} - \mathbf{u}_{j+1\langle}^{(\frac{1}{2})} + \mathbf{v}_{j+1}^0$, $\boldsymbol{\lambda}_j^N = -A_{j\langle} (\mathbf{u}_{j+1\langle}^{(1)} - \mathbf{u}_{j+1\langle}^{(\frac{1}{2})} + \mathbf{v}_{j+1}^0) - A_{j\langle j+1\bullet} (\mathbf{u}_{j+1\bullet}^{(1)} - \mathbf{u}_{j+1\bullet}^{(\frac{1}{2})} + \mathbf{v}_{j+1\bullet}^0)$. Therefore, we have $\mathbf{v}_{j\bullet} = \mathbf{u}_{j\bullet}^{(1)}$.*

Proof. From the algorithms, we have $\mathbf{u}_1^{(\frac{1}{2})} = \mathbf{v}_1^0$ and so

$$(87) \quad \boldsymbol{\lambda}_{2\langle}^D = \mathbf{u}_{1\langle}^{(\frac{1}{2})}, \quad \boldsymbol{\lambda}_{2\langle}^N = -(A_{2\langle} \mathbf{u}_{1\langle}^{(\frac{1}{2})} + A_{2\langle 1\bullet} \mathbf{u}_{1\bullet}^{(\frac{1}{2})}).$$

Since $G^{(j)}$ is the inverse of the coefficient matrix of the j th subproblem of Algorithm 2, the solution can be represented as

$$(88) \quad \begin{bmatrix} \mathbf{u}_{j\bullet}^{(\frac{1}{2})} \\ \mathbf{u}_{j\langle}^{(\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} G_{j\bullet\langle}^{(j)} & G_{j\bullet\bullet}^{(j)} & G_{j\bullet j}^{(j)} \\ G_{j\langle j\langle}^{(j)} & G_{j\langle j\bullet}^{(j)} & G_{j\langle j}^{(j)} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{j\langle} + (\tilde{S}_{j\langle} - A_{j\langle}) \mathbf{u}_{j-1\langle}^{(\frac{1}{2})} - A_{j\langle j-1\bullet} \mathbf{u}_{j-1\bullet}^{(\frac{1}{2})} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\langle} \end{bmatrix}.$$

²³In the full paper [179] that appeared after [180], the matrix form of Algorithm 13 is derived by a first-order finite difference discretization of the Neumann derivatives. $\boldsymbol{\lambda}_{j*}^N$ is not introduced but is replaced with the d.o.f. immediately next to $\boldsymbol{\lambda}_{j*}^D$ in Ω_j . Since this difference is only minor, we will not study this variant further here. A referee pointed out to us that our presentation here is closer to [178, Appendix C].

Algorithm 14. Polarized traces preconditioner at the matrix level.

Input the r.h.s. \mathbf{f} . Suppose the decomposition is nonoverlapping.

Independently for $j = 1, \dots, J$ solve

$$\begin{bmatrix} \tilde{S}_{j\langle}^{\langle} & A_{j\langle j\bullet} & \\ A_{j\bullet j\langle} & A_{j\bullet} & A_{j\bullet j\bullet} \\ & A_{j\bullet j\bullet} & \tilde{S}_{j\bullet}^{\rangle} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{j\langle}^0 \\ \mathbf{v}_{j\bullet}^0 \\ \mathbf{v}_{j\bullet}^0 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{j\langle} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\bullet} \end{bmatrix},$$

where $\tilde{S}_{j\langle}^{\langle}$ and $\tilde{S}_{j\bullet}^{\rangle}$ are defined by the Schur complements of the PML exterior to Ω_j (see Remark 14 for practical implementation). Denote by $G^{(j)}$ the inverse of the above coefficient matrix and let it be partitioned in the same way as the above coefficient matrix, e.g., $G_{j\bullet j\bullet}^{(j)} := I_{j\bullet}^{(j)} G^{(j)} I_{j\bullet}^{(j)}$. We can represent $\mathbf{v}_j^0 = G^{(j)}[\mathbf{f}_{j\langle}; \mathbf{f}_{j\bullet}; \mathbf{f}_{j\bullet}]$.

Let $\lambda_{1\langle}^D \leftarrow 0$ and $\lambda_{1\langle}^N \leftarrow 0$. Compute successively for $j = 2, \dots, J$,

$$\begin{aligned} \lambda_{j\langle}^D &\leftarrow S_{j\langle} \lambda_{j-1\langle}^N - D_{j\langle} \lambda_{j-1\langle}^D + \mathbf{v}_{j-1\langle}^0, \\ \lambda_{j\langle}^N &\leftarrow D_{j\langle}^* \lambda_{j-1\langle}^N - N_{j\langle} \lambda_{j-1\langle}^D - (A_{j\langle}^{\langle} \mathbf{v}_{j-1\langle}^0 + A_{j\langle j-1\bullet} \mathbf{v}_{j-1\bullet}^0), \end{aligned}$$

where the matrix potentials vanish for $j = 2$ and are given for $j \geq 3$ by

$$S_{j\langle} := G_{j\langle j-1\langle}^{(j-1)}, \quad D_{j\langle} := G_{j\langle j-1\langle}^{(j-1)} A_{j-1\langle}^{\langle} + G_{j\langle j-1\bullet}^{(j-1)} A_{j-1\bullet j-1\langle}, \quad D_{j\langle}^* := -A_{j\langle}^{\langle} G_{j\langle j-1\langle}^{(j-1)} - A_{j\langle j-1\bullet} G_{j-1\bullet j-1\langle}^{(j-1)},$$

$$N_{j\langle} := -A_{j\langle}^{\langle} (G_{j\langle j-1\langle}^{(j-1)} A_{j-1\langle}^{\langle} + G_{j\langle j-1\bullet}^{(j-1)} A_{j-1\bullet j-1\langle}) - A_{j\langle j-1\bullet} (G_{j-1\bullet j-1\langle}^{(j-1)} A_{j-1\langle}^{\langle} + G_{j-1\bullet}^{(j-1)} A_{j-1\bullet j-1\langle}).$$

Let $\lambda_{J\langle}^D \leftarrow 0$ and $\lambda_{J\langle}^N \leftarrow 0$. Compute successively for $j = J-1, \dots, 1$,

$$\begin{aligned} \lambda_{j\langle}^D &\leftarrow S_{j\langle} \lambda_{j+1\langle}^N - D_{j\langle} \lambda_{j+1\langle}^D + \mathbf{v}_{j+1\langle}^0, \\ \lambda_{j\langle}^N &\leftarrow D_{j\langle}^* \lambda_{j+1\langle}^N - N_{j\langle} \lambda_{j+1\langle}^D - (A_{j\langle}^{\langle} \mathbf{v}_{j+1\langle}^0 + A_{j\langle j+1\bullet} \mathbf{v}_{j+1\bullet}^0), \end{aligned}$$

where the matrix potentials vanish for $j = J-1$ and are given for $j \leq J-2$ by

$$S_{j\langle} := G_{j\langle j+1\langle}^{(j+1)}, \quad D_{j\langle} := G_{j\langle j+1\langle}^{(j+1)} A_{j+1\langle}^{\langle} + G_{j\langle j+1\bullet}^{(j+1)} A_{j+1\bullet j+1\langle}, \quad D_{j\langle}^* := -A_{j\langle}^{\langle} G_{j\langle j+1\langle}^{(j+1)} - A_{j\langle j+1\bullet} G_{j+1\bullet j+1\langle}^{(j+1)},$$

$$N_{j\langle} := -A_{j\langle}^{\langle} (G_{j\langle j+1\langle}^{(j+1)} A_{j+1\langle}^{\langle} + G_{j\langle j+1\bullet}^{(j+1)} A_{j+1\bullet j+1\langle}) - A_{j\langle j+1\bullet} (G_{j+1\bullet j+1\langle}^{(j+1)} A_{j+1\langle}^{\langle} + G_{j+1\bullet}^{(j+1)} A_{j+1\bullet j+1\langle}).$$

Recover the subdomain solutions independently for $j = 1, \dots, J$,

$$\mathbf{v}_{j\bullet} \leftarrow G_{j\bullet j\langle}^{(j)} \lambda_{j\langle}^N - (G_{j\bullet j\langle}^{(j)} A_{j\langle}^{\langle} + G_{j\bullet}^{(j)} A_{j\bullet j\langle}) \lambda_{j\langle}^D + G_{j\bullet j\bullet}^{(j)} \lambda_{j\bullet}^N - (G_{j\bullet j\bullet}^{(j)} A_{j\bullet}^{\langle} + G_{j\bullet}^{(j)} A_{j\bullet j\bullet}) \lambda_{j\bullet}^D + \mathbf{v}_{j\bullet}^0.$$

$$\text{Output } \tilde{\mathbf{u}} \leftarrow \sum_{j=1}^J R_j^T (I_j^{(j)} \lambda_{j\langle}^D + I_j^{(j)} \mathbf{v}_{j\bullet}).$$

We claim that the following identity holds:

$$(89) \quad \begin{bmatrix} \mathbf{u}_{j\bullet}^{(\frac{1}{2})} \\ \mathbf{u}_{j\langle}^{(\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} G_{j\bullet j\langle}^{(j)} & G_{j\bullet}^{(j)} & G_{j\bullet j\bullet}^{(j)} \\ G_{j\langle j\langle}^{(j)} & G_{j\langle}^{(j)} & G_{j\langle}^{(j)} \\ & G_{j\langle j\bullet}^{(j)} & G_{j\langle}^{(j)} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{j\langle} \\ \mathbf{f}_{j\bullet} \\ \mathbf{f}_{j\bullet} \end{bmatrix} - \begin{bmatrix} G_{j\bullet j\langle}^{(j)} \\ G_{j\langle j\langle}^{(j)} \end{bmatrix} (A_{j\langle}^{\langle} \mathbf{u}_{j-1\langle}^{(\frac{1}{2})} + A_{j\langle j-1\bullet} \mathbf{u}_{j-1\bullet}^{(\frac{1}{2})}) \\ - \left\{ \begin{bmatrix} G_{j\bullet j\langle}^{(j)} \\ G_{j\langle j\langle}^{(j)} \end{bmatrix} A_{j\langle}^{\langle} + \begin{bmatrix} G_{j\bullet}^{(j)} & G_{j\bullet j\bullet}^{(j)} \\ G_{j\langle j\bullet}^{(j)} & G_{j\langle}^{(j)} \end{bmatrix} \begin{bmatrix} A_{j\bullet j\langle} \\ 0 \end{bmatrix} \right\} \mathbf{u}_{j-1\langle}^{(\frac{1}{2})}.$$

In fact, the difference in the r.h.s. between (88) and (89) is

$$\left\{ \begin{bmatrix} G_{j\bullet j\langle}^{(j)} & G_{j\bullet}^{(j)} \\ G_{j\rangle j\langle}^{(j)} & G_{j\rangle j\bullet}^{(j)} \end{bmatrix} \begin{bmatrix} \tilde{S}_{j\langle} \\ A_{j\bullet j\langle} \end{bmatrix} \right\} \mathbf{u}_{j-1\lceil}^{(\frac{1}{2})},$$

and the matrix in the braces vanishes because $G^{(j)}$ is the inverse of the coefficient matrix of the subproblem for \mathbf{v}_j^0 of Algorithm 14. Assuming that

$$(90) \quad \lambda_{j\langle}^D = \mathbf{u}_{j-1\lceil}^{(\frac{1}{2})}, \quad \lambda_{j\langle}^N = -A_{j\langle}^\rangle \mathbf{u}_{j-1\lceil}^{(\frac{1}{2})} - A_{j\langle j-1\bullet} \mathbf{u}_{j-1\bullet}^{(\frac{1}{2})},$$

we substitute (90) into (89), take the Dirichlet and Neumann traces on $\Gamma_{j+1,j}$, and compare the results with the updating rules in Algorithm 14 to see that (90) also holds for $j+1$ replacing j . By induction based on (87), we conclude that (90) holds for all $j = 2, \dots, J$. Substituting (90) and \mathbf{v}_j^0 into (89) yields (for $j = J$ change the l.h.s. to $\mathbf{u}_J^{(1)}$)

$$(91) \quad \mathbf{u}_{j\bullet}^{(\frac{1}{2})} = \mathbf{v}_{j\bullet}^0 + G_{j\bullet j\langle}^{(j)} \lambda_{j\langle}^N - \left(G_{j\bullet j\langle}^{(j)} A_{j\langle}^\rangle + G_{j\bullet}^{(j)} A_{j\bullet j\langle} \right) \lambda_{j\langle}^D.$$

In particular, we have $\mathbf{u}_{j\bullet}^{(1)} = \mathbf{v}_{j\bullet}$ with \mathbf{v}_J from Algorithm 14.

In the backward sweep of Algorithm 2, we denote $\mathbf{w}_j := \mathbf{u}_j^{(1)} - \mathbf{u}_j^{(\frac{1}{2})} + \mathbf{v}_j^0$, $j = J-1, \dots, 1$, and $\mathbf{w}_J := \mathbf{u}_J^{(1)}$. By arguments similar to the last paragraph we can show for all $j = J-1, \dots, 1$ that

$$\lambda_{j\rangle}^D = \mathbf{w}_{j+1\lceil}, \quad \lambda_{j\rangle}^N = -A_{j\rangle}^\langle \mathbf{w}_{j+1\lceil} - A_{j\rangle j+1\bullet} \mathbf{w}_{j+1\bullet},$$

and further

$$(92) \quad \mathbf{w}_{j\bullet} = \mathbf{v}_{j\bullet}^0 + G_{j\bullet j\rangle}^{(j)} \lambda_{j\rangle}^N - \left(G_{j\bullet j\rangle}^{(j)} A_{j\rangle}^\langle + G_{j\bullet}^{(j)} A_{j\bullet j\rangle} \right) \lambda_{j\rangle}^D.$$

Combining (91) and (92), we conclude that $\mathbf{u}_{j\bullet}^{(1)} = \mathbf{v}_{j\bullet}$. \square

Remark 43. In Algorithm 14, the global approximation $\tilde{\mathbf{u}}$ is different from $\mathbf{u}^{(1)}$ of Algorithm 4. In Algorithm 14, $R_{j\rangle} \tilde{\mathbf{u}} = \mathbf{u}_{j\rangle}^{(\frac{1}{2})}$ with $\mathbf{u}_j^{(\frac{1}{2})}$ from Algorithm 2, while Algorithm 4 takes $R_{j\rangle} \mathbf{u}^{(1)} = \mathbf{u}_{j+1\lceil}^{(1)}$ with $\mathbf{u}_{j+1}^{(1)}$ from Algorithm 2.

9. Optimal Parallel Schwarz Methods for Arbitrary Decompositions. All the methods discussed so far only apply for a domain decomposition into a sequence of subdomains, and the information is passed gradually from one subdomain to its neighbor through the linear adjacency of the decomposition. The methods converge after one double sweep if the appropriate DtN operators are used in the transmission conditions between the subdomains. If the subdomain solves are performed in parallel, then the methods converge in a number of iterations that equals the number of subdomains, as was first pointed out in [133] (see also [131]), and this result was generalized

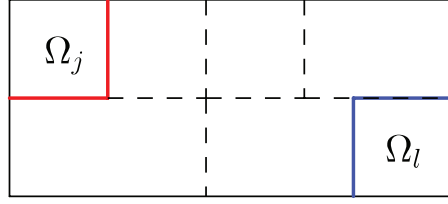


Fig. 5 Two subdomains in an arbitrary decomposition.

in [135] to domain decompositions whose connectivity graph has no cycles. Whether an optimal Schwarz method exists for an *arbitrary* decomposition was a question until the method was first created in [72]. The method converges in two iterations and thus the iteration matrix is nilpotent of degree two; each iteration exposes parallelism *between* the subdomains in solving the subproblems and after the first iteration an all-to-all communication is invoked to transmit the interface data between every pair of subdomains (even if they are not adjacent). Note that the communication happens on the whole interfaces of the subdomains, e.g., Ω_j will map the data on the entirety of $\partial\Omega_j \cap \Omega$ to the data on the entirety of $\partial\Omega_l \cap \Omega$ and send them to Ω_l ; see Figure 5.

The optimal algorithm in [72] was derived at the discrete level using linear algebra techniques, and is thus valid for various types of discretized PDEs. We give here an equivalent formulation at the continuous level, to complete the pair of discrete and continuous algorithms as was done for all the other methods in this review. In the optimal algorithm from [72], Ω_l needs to take into account all the exterior sources as well as the interior source. Taking into account the interior source requires putting a transparent boundary condition on $\partial\Omega_l \cap \Omega$, while the exterior sources have to be taken into account with the data $\lambda_l := \sum_{j \neq l} \lambda_{l,j}$ in the transmission condition on $\partial\Omega_l \cap \Omega$. Here, $\lambda_{l,j}$ and $\lambda_{j,j}$ are certain types of traces on $\partial\Omega_l \cap \Omega$ and $\partial\Omega_j \cap \Omega$ of the wave field generated by the source in Ω_j . The map that turns $\lambda_{j,j}$ into $\lambda_{l,j}$ is

$$(93) \quad \mathcal{F}_{l,j} : \lambda_{j,j} \rightarrow \lambda_{l,j} = (\mathcal{B}_l v)|_{\partial\Omega_l \cap \Omega}, \text{ s.t. } \begin{cases} \mathcal{L}v = 0 & \text{in } \Omega - \Omega_j, \\ \mathcal{B}v = 0 & \text{on } \partial\Omega - \partial\Omega_j, \\ \mathcal{C}_j v = \lambda_{j,j} & \text{on } \partial\Omega_j \cap \Omega, \end{cases}$$

where \mathcal{C}_j is the trace operator corresponding to $\lambda_{j,j}$, and \mathcal{B}_l is the trace operator corresponding to $\lambda_{l,j}$. For convenience, we can take $\mathcal{B}_l := \mathcal{T}_l$, where \mathcal{T}_l is a transparent boundary operator for the truncation of Ω to Ω_l , which allows us to simulate the waves generated by λ_l and the interior source together by the subproblem in Ω_l . We may use some approximation $\tilde{\mathcal{F}}_{l,j}$ of the operator in (93) to define a preconditioner, which leads to the algorithm given in Algorithm 15.

THEOREM 44. *If in Algorithm 15 $\tilde{\mathcal{F}}_{l,j} = \mathcal{F}_{l,j}$ is uniquely defined as in (93) and \mathcal{B}_j is an exact transparent boundary operator, then the preconditioner given by Algorithm 15 is exact, i.e., the output \tilde{u} is the solution of (11). This means the iteration operator is nilpotent of degree two.*

Proof. The proof is straightforward by well-posedness and linearity. \square

Algorithm 15. Optimized Schwarz preconditioner using **global transmission** conditions at the **PDE** level.

Input the source terms f and g .

Suppose the decomposition is arbitrary such that $\cup_{j=1}^J \bar{\Omega}_j = \bar{\Omega}$.

Solve the following subproblems independently for $j = 1, \dots, J$:

$$\begin{aligned} \mathcal{L} v_j^{(\frac{1}{2})} &= f && \text{in } \Omega_j, \\ \mathcal{B} v_j^{(\frac{1}{2})} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathcal{B}_j v_j^{(\frac{1}{2})} &= 0 && \text{on } \partial\Omega_j - \partial\Omega, \end{aligned}$$

where \mathcal{B}_j is an approximation of a transparent boundary operator for truncation of Ω to Ω_j . Take the trace $\lambda_{j,j} \leftarrow \mathcal{C}_j v_j^{(\frac{1}{2})}$ on $\partial\Omega_j - \partial\Omega$ and map it to $\lambda_{l,j} \leftarrow \tilde{\mathcal{F}}_{l,j} \lambda_{j,j}$ on $\partial\Omega_l - \partial\Omega$ for all $l \neq j$. Here, $\tilde{\mathcal{F}}_{l,j}$ is an approximation of $\mathcal{F}_{l,j}$ in (93).

Solve the following subproblems independently for $j = 1, \dots, J$:

$$\begin{aligned} \mathcal{L} v_j^{(1)} &= f && \text{in } \Omega_j, \\ \mathcal{B} v_j^{(1)} &= g && \text{on } \partial\Omega \cap \partial\Omega_j, \\ \mathcal{B}_j v_j^{(1)} &= \sum_{l \neq j} \lambda_{j,l} && \text{on } \partial\Omega_j - \partial\Omega. \end{aligned}$$

Output $\tilde{u} \leftarrow \sum_{j=1}^J \mathcal{E}_j(\phi_j v_j^{(1)})$ with \mathcal{E}_j the extension by zero to Ω , and $\sum_{j=1}^J \mathcal{E}_j \phi_j = 1$.

As seen from Theorem 44, approximating (93) is crucial for Algorithm 15. Essentially, this consists in approximating the off-diagonal part of the Green's function corresponding to the two interfaces. However, at the time of writing this article, no effort has been made toward a practical realization of Algorithm 15.

10. Numerical Experiments. The main goal of our article is both theoretical and formal, namely, to show that there is a common principle behind the new Helmholtz preconditioners based on sequential domain decomposition. Numerically, impressive results have been shown for these algorithms in the literature; see, for example, [29, 52, 111, 139, 157, 179]. Nevertheless, it is interesting and right that the readers of this review should also see when these new algorithms get into difficulty, which motivated us to add this section. Some of the results here have appeared in [85].

We consider the Helmholtz equation on the unit square

$$(94) \quad (\Delta + k(x)^2)u = f \quad \text{in } \Omega := (0, 1)^2,$$

with suitable boundary conditions for well-posedness. We discretize (94) by the classical five point finite difference method. We split the square sequentially in the x direction into $p = 4, 8, 16$ equal strips representing the subdomains with vertical interfaces. Each subdomain has its own constant wavenumber. For the case of four subdomains, we use the wavenumbers

$$(95) \quad k = [20 \ 20 \ 20 \ 20] + \alpha[0 \ 20 \ 10 \ -10],$$

where α is a contrast parameter, and for larger p we just repeat this structure. The

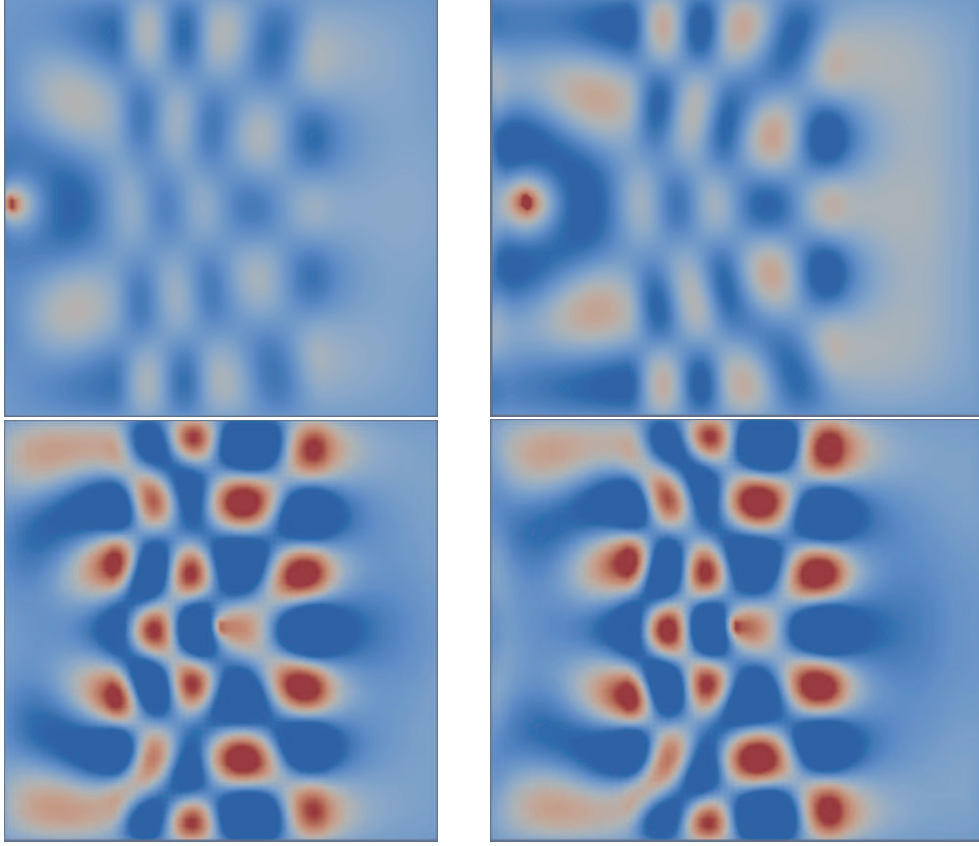


Fig. 6 Guided waves from a point source near the left boundary (first row) or near the center (second row), with homogeneous Dirichlet conditions on the top and bottom and Robin conditions (first column) or PMLs (second column) on the left and right boundaries. The media have four layers separated by vertical interfaces as in (95) with $\alpha = 1$.

mesh resolution we choose guarantees at least ten points per wavelength for this experiment. We start with the case of a wave guide in the x direction, where we use Robin or PML radiation conditions on the left and right, and homogeneous Dirichlet conditions on top and bottom. We show in Figure 6 the real part of the solution²⁴ we obtain in the four-layered medium with $\alpha = 1$ stimulated by a point source at $x = 2h$, $y = \frac{1-h}{2}$ in the top row, and for the point source at $x = \frac{1}{2}$, $y = \frac{1-h}{2}$ in the bottom row.

We are going to test the two fundamental algorithms mentioned in section 2: one based on the factorization and the other based on the nonoverlapping Schwarz method. As stated in Theorem 30, the factorization method has (approximate) DtN or Schur complement derived transmission conditions on the *left* interfaces of subdomains, and Dirichlet conditions on the right. For the nonoverlapping DOSM (see Algorithms 3 and 4), we use the version with (approximate) DtN derived transmission conditions on *both* interfaces of each subdomain. For the case of a constant medium, i.e., $\alpha = 0$ in

²⁴The Dirichlet boundary points are not plotted, but the PMLs are plotted.

Table 2 *LU iteration numbers in the wave guide setting.*

α	$p = 4$			$p = 8$			$p = 16$		
	Iterative	GMRES		Iterative	GMRES		Iterative	GMRES	
0	1	1	1	1	1	1	1	1	1
0.001	4	3	3	5	3	3	6	3	3
0.005	6	4	4	12	5	5	13	5	5
0.01	8	5	4	16	6	5	38	7	7
0.05	-	8	6	-	17	12	-	12	26
0.1	32	10	11	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-
0	1	1	1	1	1	1	1	1	1
0.001	4	2	3	5	3	3	7	4	4
0.005	7	3	3	-	5	4	30	6	6
0.01	11	4	4	-	6	5	-	10	10
0.05	-	-	-	-	-	-	-	-	-
0.1	-	22	31	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-

(95), we use the exact DtN for (94). That is, we calculate the exact Schur complement for the discretized problem. For $\alpha > 0$, we approximate the DtN for (94) by the exact DtN for the Helmholtz equation in a modified medium: when calculating the exterior DtN on the *left* interface of a subdomain, all the medium to the *left* of the subdomain is assumed to be the identity extension of the medium in the *left* neighborhood of the subdomain.²⁵ On the right interface of a subdomain we do the same. Of course, this approximation has errors for heterogeneous media, and we will see that these errors have a dramatic impact on the convergence of the algorithms, something which is not yet well documented in the literature on these methods.

We test the two algorithms both as iterative solvers and as preconditioners for GMRES for varying contrast parameter α and subdomain numbers. We do this for mesh size $h = 1/64$ and the contrast profile in (95), and on a refined mesh with $h = 1/128$, but also for a profile with twice the size for the wavenumber, i.e.,

$$(96) \quad k = [40 \ 40 \ 40 \ 40] + \alpha[0 \ 40 \ 20 \ -20],$$

so that we still have at least ten points per wavelength resolution. We show in Tables 2 and 3 the number of iterations the methods took, where we stopped the iterative version of the algorithms and GMRES when the residual was reduced by $1e-6$, and we started with a zero initial guess of the solution u for a random source term f in the physical domain Ω of (94). In the PMLs outside Ω , both the initial guess and the source term are set to zero. The three columns within each “Iterative” or “GMRES” column correspond to Robin conditions, PMLs of thickness five times the mesh size, and PMLs of thickness ten times the mesh size on the left and right of the original domain. The top parts are for the smaller wavenumber experiment in (95), and the bottom parts are for the larger wavenumber experiment in (96). We first see that for $\alpha = 0$, i.e., in the constant wavenumber case, the factorization is exact, and both the iterative version and GMRES converge in one iteration step. As soon as we have a nonconstant wavenumber, however, already for $\alpha = 0.001$, the factorization is not exact any more. Nevertheless the algorithms still converge well, up to $\alpha = 0.01$ in the

²⁵This is a common assumption upon which most of the ABC and PML techniques are based.

Table 3 Schwarz iteration numbers in the wave guide setting.

α	$p = 4$			$p = 8$			$p = 16$		
	Iterative	GMRES		Iterative	GMRES		Iterative	GMRES	
0	1	1	1	1	1	1	1	1	1
0.001	3	2	2	3	2	2	3	2	2
0.005	5	3	3	5	3	3	5	3	3
0.01	7	4	3	7	4	4	7	4	3
0.05	42	12	7	7	5	4	-	21	17
0.1	-	-	-	9	7	6	-	-	-
1	-	-	-	-	-	-	14	12	11
							-	-	-
							48	47	47
							-	-	-
							59	68	65

0	1	1	1	1	1	1	1	1	1
0.001	3	2	3	3	2	3	3	2	3
0.005	5	3	4	4	3	3	5	3	3
0.01	8	4	6	5	4	4	6	4	4
0.05	-	-	-	10	7	6	-	-	-
0.1	-	-	-	13	11	9	-	-	-
1	-	-	-	43	40	38	-	-	-

smaller wavenumber case in the top parts of the tables, i.e., a one percent variation in the wavenumber k . For larger contrast, the iterative version of the algorithms cannot be used any more, and GMRES now deteriorates rapidly, for example, if the contrast is at a factor of two, i.e., $\alpha = 1$, GMRES iteration numbers double when p goes from 4 to 8, and the two algorithms are not robust any more. In the higher wavenumber case in the bottom parts of the tables, they deteriorate even more rapidly for higher contrast. We can also see by comparing the last lines in the top and bottom parts of the tables that doubling the wavenumber leads to a remarkable growth of the iteration numbers with GMRES as soon as the contrast is large enough, and that GMRES fails to converge in less than hundred iterations at the bottom right.

We next perform the same set of experiments, but now using Robin or PML conditions all around the original domain; see Figure 7, Table 4, and Table 5. We see that the outer radiation conditions are better than the wave guide setting for the two algorithms, and they now work in the iterative version up to about a 10 percent variation of the wavenumber in this specific experiment. As soon as there is a variation as large as a factor of two, however, the algorithms are not effective solvers any more, the iterative versions diverge, and GMRES iteration numbers deteriorate when the number of subdomains increases and also when the wavenumber is doubled. One thus has to be careful when claiming optimality of algorithms in this class of Helmholtz preconditioners.

II. Conclusions. We have seen that for a large class of new Helmholtz solvers the underlying mathematical technique is the same: the solvers are based on a nilpotent iteration given by a double sweep in a sequential domain decomposition that uses the exact Dirichlet-to-Neumann operators for transmission conditions at the interfaces. At the linear algebra level, the corresponding algorithm is based on an exact block LU factorization of the matrix. From domain decomposition, it is known that when the solves are performed in parallel, instead of in a sweeping fashion, the method is still nilpotent, but convergence is then achieved in a number of iterations corresponding to the number of subdomains [131, 133]. If the domain decomposition is more general, such that the connectivity graph includes cycles and thus cross points between subdomains are present, we have given an algorithm at the continuous level based on the discrete algorithm in [72] that is still nilpotent. This algorithm requires

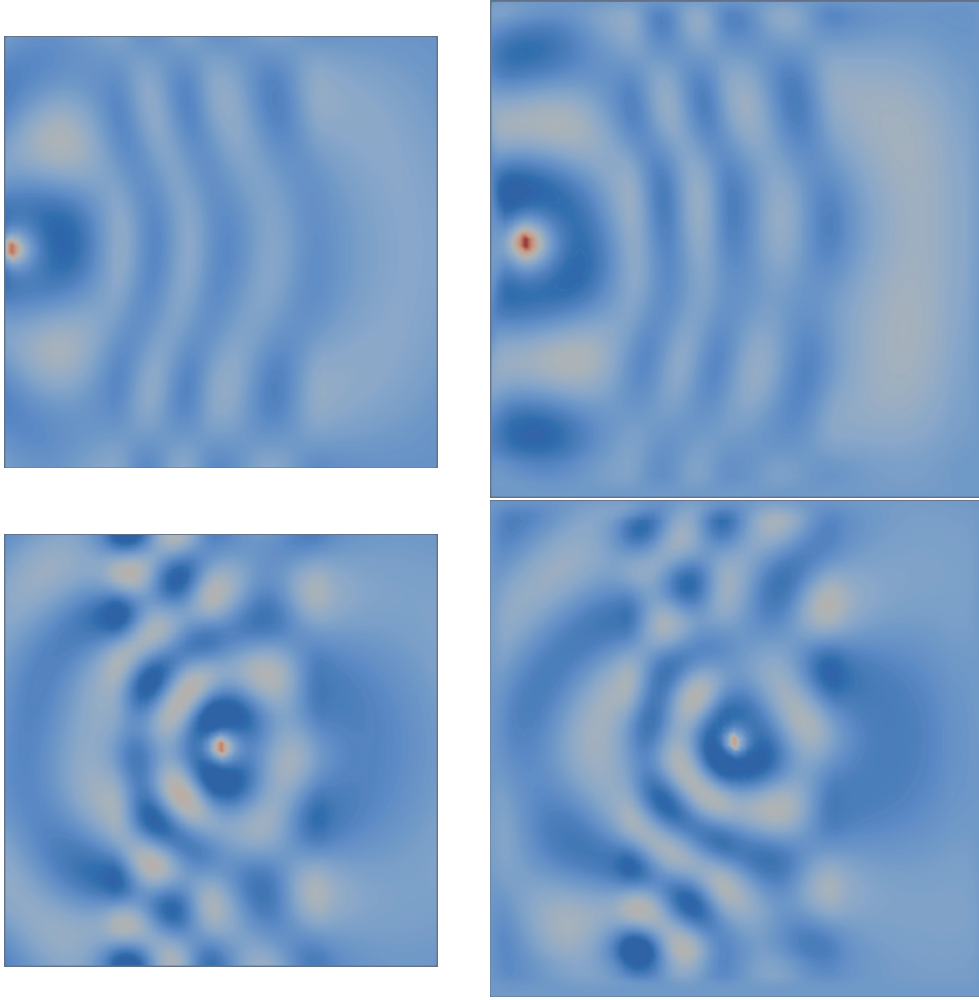


Fig. 7 *Scattered waves from a point source near the left boundary (first row) or near the center (second row), with Robin conditions (first column) or PMLs (second column, bigger since the plot includes the PML region) all around the square domain. The media have four layers separated by vertical interfaces as in (95) with $\alpha = 1$.*

communication of each subdomain with every other one, and convergence is achieved in two iterations. While there is currently no practical realization of this algorithm, the fact that the algorithm converges in two iterations, independent of the number of subdomains, suggests that a coarse space component is active in this optimal algorithm. Coarse spaces leading to nilpotent iterations were first described in the lecture notes [68], and then in [70, 71], with successful approximations in [74, 75]. The property that domain decomposition methods in general are nilpotent has only very recently been investigated in more detail; see [23]. None of the Helmholtz solvers we described in this article use coarse space techniques at the time of writing, which leads us to expect that this area of research will remain very active over the coming years.

Table 4 *LU iteration numbers for a domain with Robin or PMLs all around.*

	$p = 4$						$p = 8$						$p = 16$					
α	Iterative			GMRES			Iterative			GMRES			Iterative			GMRES		
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.001	2	2	2	2	2	2	3	3	2	3	2	2	3	3	3	3	3	3
0.005	3	3	3	3	3	3	4	3	3	4	3	3	4	3	3	4	3	3
0.01	3	3	3	3	3	3	4	4	3	4	3	3	5	4	4	5	4	3
0.05	5	5	5	5	4	4	7	5	5	6	5	5	8	6	5	8	5	5
0.1	7	6	5	6	5	5	9	6	6	8	6	6	12	7	7	10	7	6
1	-	31	27	15	12	12	-	-	-	32	25	22	-	-	-	58	34	29
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.001	3	2	2	3	2	2	3	2	2	3	2	2	3	3	3	3	3	3
0.005	3	3	3	3	3	3	4	3	3	4	3	3	5	3	3	5	3	3
0.01	4	3	3	4	3	3	5	3	3	5	3	3	6	4	4	6	4	4
0.05	6	4	4	6	4	4	10	5	5	8	5	5	14	6	6	11	6	6
0.1	8	5	5	7	5	5	13	6	6	10	6	6	14	8	7	12	8	7
1	-	-	52	23	13	12	-	-	-	39	24	22	-	-	-	99	61	48

Table 5 Schwarz iteration numbers for a domain with Robin conditions or PMLs all around.

	$p = 4$						$p = 8$						$p = 16$					
α	Iterative			GMRES			Iterative			GMRES			Iterative			GMRES		
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.001	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0.005	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0.01	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0.05	5	4	4	4	4	4	5	5	4	4	4	4	5	5	4	4	4	4
0.1	6	5	5	5	5	4	6	6	5	5	5	5	7	6	5	6	5	5
1	-	-	-	23	33	37	-	-	-	35	44	44	-	-	-	41	43	48
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.001	3	2	2	2	2	2	3	3	2	2	2	2	3	3	2	2	2	2
0.005	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0.01	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0.05	5	5	5	5	4	4	5	5	5	5	4	4	5	5	5	5	5	4
0.1	7	7	6	6	6	5	9	7	6	7	6	5	9	7	7	7	6	6
1	-	-	-	32	43	45	-	-	-	49	54	61	-	-	-	79	94	75

REFERENCES

- [1] Y. ACHDOU AND F. NATAF, *Dimension-wise iterated frequency filtering decomposition*, Numer. Linear Algebra Appl., 41 (2003), pp. 1643–1681. (Cited on p. 14)
- [2] P. R. AMESTOY, I. S. DUFF, J.-Y. L’EXCELLENT, AND J. KOSTER, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41, <https://doi.org/10.1137/S0895479899358194>. (Cited on p. 11)
- [3] X. ANTOINE AND M. DARBAS, *Integral equations and iterative schemes for acoustic scattering problems*, in Numerical Methods for Acoustics Problems, F. Magoulès, ed., Saxe-Coburg Publications, 2016, Chapter 2. (Cited on p. 12)
- [4] D. N. ARNOLD, F. BREZZI, B. COCKBURN, AND L. D. MARINI, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., 39 (2002), pp. 1749–1779, <https://doi.org/10.1137/S0036142901384162>. (Cited on p. 17)
- [5] I. M. BABUŠKA AND S. A. SAUTER, *Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?*, SIAM J. Numer. Anal., 34 (1997), pp. 2392–2423, <https://doi.org/10.1137/S0036142994269186>. (Cited on p. 4)
- [6] H. BAGCI, J. E. PASCIAK, AND K. Y. SIRENKO, *A convergence analysis for a sweeping preconditioner for block tridiagonal systems of linear equations*, Numer. Linear Algebra Appl., 22 (2015), pp. 371–392. (Cited on p. 14)
- [7] A. BAMBERGER, P. JOLY, AND J. E. ROBERTS, *Second-order absorbing boundary conditions for the wave equation: A solution for the corner problem*, SIAM J. Numer. Anal., 27 (1990), pp. 323–352, <https://doi.org/10.1137/0727021>. (Cited on p. 10)

- [8] L. BANJAI AND W. HACKBUSCH, *Hierarchical matrix techniques for low- and high-frequency Helmholtz problems*, IMA J. Numer. Anal., 28 (2008), pp. 46–79. (Cited on pp. 11, 12)
- [9] M. BEBENDORF, *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, Springer-Verlag, Berlin, 2008. (Cited on p. 11)
- [10] M. BEBENDORF AND W. HACKBUSCH, *Existence of \mathcal{H} -matrix approximants to the inverse FE-matrix of elliptic operators with L^∞ coefficients*, Numer. Math., 95 (2003), pp. 1–28. (Cited on p. 11)
- [11] M. BEBENDORF, C. KUSKE, AND R. VENN, *Wideband nested cross approximation for Helmholtz problems*, Numer. Math., 130 (2015), pp. 1–34. (Cited on p. 11)
- [12] A. BENDALI AND Y. BOUBENDIR, *Non-overlapping domain decomposition method for a nodal finite element method*, Numer. Math., 103 (2006), pp. 515–537. (Cited on p. 33)
- [13] J.-P. BERENGER, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp. 185–200. (Cited on p. 8)
- [14] M. BOLLHÖFER, M. J. GROTE, AND O. SCHENK, *Algebraic multilevel preconditioner for the Helmholtz equation in heterogeneous media*, SIAM J. Sci. Comput., 31 (2009), pp. 3781–3805, <https://doi.org/10.1137/080725702>. (Cited on p. 12)
- [15] Y. BOUBENDIR, X. ANTOINE, AND C. GEUZAINÉ, *A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation*, J. Comput. Phys., 231 (2012), pp. 262–280. (Cited on p. 13)
- [16] A. BRANDT AND I. LIVSHITS, *Wave-ray multigrid method for standing wave equations*, Electron. Trans. Numer. Anal., 6 (1997), pp. 162–181. (Cited on p. 13)
- [17] X.-C. CAI, M. A. CASARIN, F. W. ELLIOTT, JR., AND O. B. WIDLUND, *Overlapping Schwarz algorithms for solving Helmholtz’s equation*, in Domain Decomposition Methods 10, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, 1998, pp. 391–398. (Cited on p. 13)
- [18] X.-C. CAI AND M. SARKIS, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797, <https://doi.org/10.1137/S106482759732678X>. (Cited on p. 30)
- [19] X.-C. CAI AND O. B. WIDLUND, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 243–258, <https://doi.org/10.1137/0913013>. (Cited on pp. 12, 13)
- [20] H. CALANDRA, S. GRATTON, X. PINEL, AND X. VASSEUR, *An improved two-grid preconditioner for the solution of three-dimensional Helmholtz problems in heterogeneous media*, Numer. Linear Algebra Appl., 20 (2012), pp. 663–688. (Cited on p. 12)
- [21] S. N. CHANDLER-WILDE, I. G. GRAHAM, S. LANGDON, AND E. A. SPENCE, *Numerical-asymptotic boundary integral methods in high-frequency acoustic scattering*, Acta Numer., 21 (2012), pp. 89–305. (Cited on p. 4)
- [22] S. CHANDRASEKARAN, P. DEWILDE, M. GU, W. LYONS, AND T. PALS, *A fast solver for HSS representations via sparse matrices*, SIAM J. Matrix Anal., 29 (2006), pp. 67–81, <https://doi.org/10.1137/050639028>. (Cited on p. 11)
- [23] F. CHAOUQUI, M. J. GANDER, AND K. SANTUGINI-REPIQUET, *On nilpotent subdomain iterations*, in Domain Decomposition Methods in Science and Engineering XXIII, C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park, and O. B. Widlund, eds., Springer International, 2017, pp. 125–133. (Cited on p. 67)
- [24] H. CHEN, P. LU, AND X. XU, *A robust multilevel method for hybridizable discontinuous Galerkin method for the Helmholtz equation*, J. Comput. Phys., 264 (2014), pp. 133–151. (Cited on p. 13)
- [25] W. CHEN, Y. LIU, AND X. XU, *A robust domain decomposition method for the Helmholtz equation with high wave number*, ESAIM Math. Model. Numer. Anal., 50 (2016), pp. 921–944. (Cited on p. 14)
- [26] Z. CHEN, D. CHENG, W. FENG, T. WU, AND H. YANG, *A multigrid-based preconditioned Krylov subspace method for the Helmholtz equation with PML*, J. Math. Anal. Appl., 383 (2011), pp. 522–540. (Cited on p. 12)
- [27] Z. CHEN, M. J. GANDER, AND H. ZHANG, *On the relation between optimized Schwarz methods and source transfer*, in Domain Decomposition Methods in Science and Engineering XXII, T. Dickopf, M. J. Gander, L. Halpern, R. Krause, and L. F. Pavarino, eds., Springer International, 2016, pp. 217–225. (Cited on p. 14)
- [28] Z. CHEN AND X. XIANG, *A source transfer domain decomposition method for Helmholtz equations in unbounded domain*, SIAM J. Numer. Anal., 51 (2013), pp. 2331–2356, <https://doi.org/10.1137/130917144>. (Cited on pp. 14, 31, 43, 44, 46)
- [29] Z. CHEN AND X. XIANG, *A source transfer domain decomposition method for Helmholtz equations in unbounded domain, Part II: Extensions*, Numer. Math. Theory Methods Appl., 6 (2013), pp. 538–555. (Cited on pp. 14, 63)

- [30] P. CHEVALIER AND F. NATAF, *Symmetrized method with optimized second-order conditions for the Helmholtz equation*, in Domain Decomposition Methods 10, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, 1998, pp. 400–407. (Cited on pp. 13, 33)
- [31] W.-C. CHEW, J.-M. JIN, AND E. MICHELSEN, *Complex coordinate stretching as a generalized absorbing boundary condition*, Microw. Opt. Technol. Lett., 15 (1997), pp. 363–369. (Cited on p. 8)
- [32] P.-H. COCQUET AND M. J. GANDER, *How large a shift is needed in the shifted Helmholtz preconditioner for its effective inversion by multigrid?*, SIAM J. Sci. Comput., 39 (2017), pp. A438–A478, <https://doi.org/10.1137/15M102085X>. (Cited on p. 12)
- [33] F. COLLINO, *High order absorbing boundary conditions for wave propagation models: Straight line boundary and corner cases*, in Proceedings of the Second International Conference on Mathematical and Numerical Aspects of Wave Propagation, R. Kleinman et al., eds., SIAM, 1993, pp. 161–171. (Cited on p. 10)
- [34] F. COLLINO, S. GHANEMI, AND P. JOLY, *Domain decomposition method for harmonic wave propagation: A general presentation*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 171–211. (Cited on p. 13)
- [35] L. CONEN, V. DOLEAN, R. KRAUSE, AND F. NATAF, *A coarse space for heterogeneous Helmholtz problems based on the Dirichlet-to-Neumann operator*, J. Comput. Appl. Math., 271 (2014), pp. 83–99. (Cited on p. 13)
- [36] S. COOLS, B. REPS, AND W. VANROOSE, *A new level-dependent coarse grid correction scheme for indefinite Helmholtz problems*, Numer. Linear Algebra Appl., 21 (2014), pp. 513–533. (Cited on p. 12)
- [37] S. COOLS AND W. VANROOSE, *Local Fourier analysis of the complex shifted Laplacian preconditioner for Helmholtz problems*, Numer. Linear Algebra Appl., 20 (2013), pp. 575–597. (Cited on p. 12)
- [38] T. A. DAVIS, *Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 196–199. (Cited on p. 11)
- [39] T. A. DAVIS, *Direct Methods for Sparse Linear Systems*, SIAM, 2006, <https://doi.org/10.1137/1.9780898718881>. (Cited on p. 11)
- [40] A. DE LA BOURDONNAYE, C. FARHAT, A. MACEDO, F. MAGOULÈS, AND F.-X. ROUX, *A non-overlapping domain decomposition method for the exterior Helmholtz problem*, in Domain Decomposition Methods 10, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, 1998, pp. 42–66. (Cited on p. 13)
- [41] B. DESPRÉS, *Méthodes de Décomposition de Domaine pour les Problèmes de Propagation d'Ondes en Régime Harmonique*, Ph.D. thesis, Université Dauphine, Paris, 1991. (Cited on p. 13)
- [42] V. DOLEAN, P. JOLIVET, AND F. NATAF, *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*, SIAM, 2015, <https://doi.org/10.1137/1.9781611974065>. (Cited on p. 13)
- [43] J. DOUGLAS, JR. AND D. B. MEADE, *Second-order transmission conditions for the Helmholtz equation*, in Ninth International Conference on Domain Decomposition Methods, P. E. Björstad, M. S. Espedal, and D. E. Keyes, eds., DDM.org, 1998, pp. 434–441. (Cited on p. 13)
- [44] V. DRUSKIN, S. GÜTTEL, AND L. KNIZHNERMAN, *Near-optimal perfectly matched layers for indefinite Helmholtz problems*, SIAM Rev., 58 (2016), pp. 90–116, <https://doi.org/10.1137/140966927>. (Cited on p. 8)
- [45] V. DRUSKIN AND L. KNIZHNERMAN, *Gaussian spectral rules for the three-point second differences: I. A two-point positive definite problem in a semi-infinite domain*, SIAM J. Numer. Anal., 37 (1999), pp. 403–422, <https://doi.org/10.1137/S0036142997330792>. (Cited on p. 10)
- [46] M. DRYJA AND O. B. WIDLUND, *An Additive Variant of the Schwarz Alternating Method for the Case of Many Subregions*, Tech. Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987. (Cited on p. 28)
- [47] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Oxford University Press, 1986. (Cited on p. 11)
- [48] E. EFSTATHIOU AND M. J. GANDER, *Why restricted additive Schwarz converges faster than additive Schwarz*, BIT, 43 (2003), pp. 945–959. (Cited on p. 28)
- [49] B. ENGQUIST AND A. MAJDA, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp., 31 (1977), pp. 629–651. (Cited on p. 8)
- [50] B. ENGQUIST AND A. MAJDA, *Radiation boundary conditions for acoustic and elastic wave calculations*, Comm. Pure Appl. Math., 32 (1979), pp. 313–357. (Cited on p. 8)

- [51] B. ENGQUIST AND L. YING, *Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation*, Comm. Pure Appl. Math., 64 (2011), pp. 697–735. (Cited on pp. 11, 12, 14, 23, 38)
- [52] B. ENGQUIST AND L. YING, *Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers*, Multiscale Model. Simul., 9 (2011), pp. 686–710, <https://doi.org/10.1137/100804644>. (Cited on pp. 14, 23, 38, 63)
- [53] B. ENGQUIST AND H. ZHAO, *Absorbing boundary conditions for domain decomposition*, Appl. Numer. Math., 27 (1998), pp. 341–365. (Cited on pp. 7, 13)
- [54] B. ENGQUIST AND H. ZHAO, *Approximate separability of Green’s function of the Helmholtz equation in the high-frequency limit*, Comm. Pure Appl. Math., 71 (2018), pp. 2220–2274. (Cited on p. 11)
- [55] Y. A. ERLANGGA, *Advances in iterative methods and preconditioners for the Helmholtz equation*, Arch. Comput. Methods Eng., 15 (2008), pp. 37–66. (Cited on pp. 4, 12)
- [56] Y. A. ERLANGGA AND R. NABBEN, *On a multilevel Krylov method for the Helmholtz equation preconditioned by shifted Laplacian*, Electron. Trans. Numer. Anal., 31 (2008), pp. 403–424. (Cited on p. 13)
- [57] Y. A. ERLANGGA, C. W. OOSTERLEE, AND C. VUIK, *A novel multigrid based preconditioner for heterogeneous Helmholtz problems*, SIAM J. Sci. Comput., 27 (2006), pp. 1471–1492, <https://doi.org/10.1137/040615195>. (Cited on p. 12)
- [58] Y. A. ERLANGGA, C. VUIK, AND C. W. OOSTERLEE, *On a class of preconditioners for the Helmholtz equation*, Appl. Numer. Math., 50 (2004), pp. 409–425. (Cited on p. 12)
- [59] O. ERNST AND M. J. GANDER, *Why it is difficult to solve Helmholtz problems with classical iterative methods*, in Numerical Analysis of Multiscale Problems, I. Graham, T. Hou, O. Lakkis, and R. Scheichl, eds., Springer-Verlag, 2012, pp. 325–363. (Cited on pp. 4, 12)
- [60] M. ESLAMINIA AND M. N. GUDDATI, *A double-sweeping preconditioner for the Helmholtz equation*, J. Comput. Phys., 314 (2016), pp. 800–823. (Cited on p. 14)
- [61] C. FARHAT, P. AVERY, R. TEZAUR, AND J. LI, *FETI-DPH: A dual-primal domain decomposition method for acoustic scattering*, J. Comput. Acoust., 13 (2005), pp. 499–524. (Cited on p. 13)
- [62] C. FARHAT, A. MACEDO, M. LESOINNE, F.-X. ROUX, F. MAGOULÈS, AND A. D. L. BOURDONNAIE, *Two-level domain decomposition methods with Lagrange multipliers for the fast iterative solution of acoustic scattering problems*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 213–239. (Cited on pp. 13, 33)
- [63] C. FARHAT AND F.-X. ROUX, *Implicit parallel processing in structural mechanics*, Comput. Mech. Adv., 2 (1994), pp. 1–124. (Cited on p. 13)
- [64] M. J. GANDER, *Optimized Schwarz methods*, SIAM J. Numer. Anal., 44 (2006), pp. 699–731, <https://doi.org/10.1137/S0036142903425409>. (Cited on p. 13)
- [65] M. J. GANDER, *Schwarz methods over the course of time*, Electron. Trans. Numer. Anal., 31 (2008), pp. 228–255. (Cited on p. 28)
- [66] M. J. GANDER, I. G. GRAHAM, AND E. A. SPENCE, *Applying GMRES to the Helmholtz equation with shifted Laplacian preconditioning: What is the largest shift for which wavenumber-independent convergence is guaranteed?*, Numer. Math., 131 (2015), pp. 567–614. (Cited on p. 12)
- [67] M. J. GANDER AND S. HAJIAN, *Analysis of Schwarz methods for a hybridizable discontinuous Galerkin discretization*, SIAM J. Numer. Anal., 53 (2015), pp. 573–597, <https://doi.org/10.1137/140961857>. (Cited on p. 13)
- [68] M. J. GANDER AND L. HALPERN, *Méthodes de décomposition de domaine*, Encyclopédie électronique pour les ingénieurs, 2012. (Cited on p. 67)
- [69] M. J. GANDER, L. HALPERN, AND F. MAGOULÈS, *An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation*, Internat. J. Numer. Methods Fluids, 55 (2007), pp. 163–175. (Cited on p. 13)
- [70] M. J. GANDER, L. HALPERN, AND K. SANTUGINI-REPIQUET, *Discontinuous coarse spaces for DD-methods with discontinuous iterates*, in Domain Decomposition Methods in Science and Engineering XXI, J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi, and O. Widlund, eds., Springer, 2014, pp. 607–615. (Cited on p. 67)
- [71] M. J. GANDER, L. HALPERN, AND K. SANTUGINI-REPIQUET, *A new coarse grid correction for RAS/AS*, in Domain Decomposition Methods in Science and Engineering XXI, J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi, and O. Widlund, eds., Springer, 2014, pp. 275–283. (Cited on pp. 54, 67)
- [72] M. J. GANDER AND F. KWOK, *Optimal interface conditions for an arbitrary decomposition into subdomains*, in Domain Decomposition Methods in Science and Engineering XIX,

- Y. Huang, R. Kornhuber, O. Widlund, and J. Xu, eds., Heidelberg, 2010, Springer-Verlag, pp. 101–108. (Cited on pp. 62, 66)
- [73] M. J. GANDER AND F. KWOK, *Best Robin parameters for optimized Schwarz methods at cross points*, SIAM J. Sci. Comput., 34 (2012), pp. A1849–A1879, <https://doi.org/10.1137/110837218>. (Cited on p. 33)
- [74] M. J. GANDER AND A. LONELAND, *SHEM: An optimal coarse space for RAS and its multiscale approximation*, in Domain Decomposition Methods in Science and Engineering XXIII, C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park, and O. B. Widlund, eds., Springer International, 2017, pp. 313–321. (Cited on p. 67)
- [75] M. J. GANDER, A. LONELAND, AND T. RAHMAN, *Analysis of a New Harmonically Enriched Multiscale Coarse Space for Domain Decomposition Methods*, preprint, <https://arxiv.org/abs/1512.05285>, 2015. (Cited on p. 67)
- [76] M. J. GANDER, F. MAGOULÈS, AND F. NATAF, *Optimized Schwarz methods without overlap for the Helmholtz equation*, SIAM J. Sci. Comput., 24 (2002), pp. 38–60, <https://doi.org/10.1137/S1064827501387012>. (Cited on pp. 13, 33)
- [77] M. J. GANDER AND F. NATAF, *AILU: A preconditioner based on the analytic factorization of the elliptic operator*, Numer. Linear Algebra Appl., 7 (2000), pp. 505–526. (Cited on pp. 14, 38)
- [78] M. J. GANDER AND F. NATAF, *An incomplete LU preconditioner for problems in acoustics*, J. Comput. Acoust., 13 (2005), pp. 455–476. (Cited on pp. 14, 38)
- [79] M. J. GANDER AND Y. XU, *Optimized Schwarz methods for circular domain decompositions with overlap*, SIAM J. Numer. Anal., 52 (2014), pp. 1981–2004, <https://doi.org/10.1137/130946125>. (Cited on p. 13)
- [80] M. J. GANDER AND Y. XU, *Optimized Schwarz methods with nonoverlapping circular domain decomposition*, Math. Comp., 86 (2016), pp. 637–660. (Cited on p. 13)
- [81] M. J. GANDER AND H. ZHANG, *Domain decomposition methods for the Helmholtz equation: A numerical investigation*, in Domain Decomposition Methods in Science and Engineering XX, R. Bank, M. Holst, O. Widlund, and J. Xu, eds., Springer-Verlag, 2013, pp. 215–222. (Cited on pp. 13, 33)
- [82] M. J. GANDER AND H. ZHANG, *Algorithmic perspective of PML transmission conditions for domain decomposition methods*, in 2014 IEEE Conference on Antenna Measurements Applications, 2014, <https://doi.org/10.1109/CAMA.2014.7003360>. (Cited on p. 14)
- [83] M. J. GANDER AND H. ZHANG, *Optimized Schwarz methods with overlap for the Helmholtz equation*, in Domain Decomposition Methods in Science and Engineering XXI, J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi, and O. Widlund, eds., Springer-Verlag, 2014, pp. 207–214. (Cited on p. 13)
- [84] M. J. GANDER AND H. ZHANG, *Optimized Schwarz methods with overlap for the Helmholtz equation*, SIAM J. Sci. Comput., 38 (2016), pp. A3195–A3219, <https://doi.org/10.1137/15M1021659>. (Cited on p. 13)
- [85] M. J. GANDER AND H. ZHANG, *Restrictions on the use of sweeping type preconditioners for Helmholtz problems*, in Domain Decomposition Methods in Science and Engineering XXIV, P. E. Bjorstad, S. C. Brenner, L. Halpern, R. Kornhuber, H. H. Kim, T. Rahman, and O. B. Widlund, eds., Springer, 2018, pp. 325–336. (Cited on p. 63)
- [86] A. GEORGE, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363, <https://doi.org/10.1137/0710032>. (Cited on p. 11)
- [87] S. GHANEMI, *A domain decomposition method for Helmholtz scattering problems*, in Ninth International Conference on Domain Decomposition Methods, P. E. Bjorstad, M. S. Espedal, and D. E. Keyes, eds., DDM.org, 1998, pp. 105–112. (Cited on p. 13)
- [88] A. GILLMAN AND P.-G. MARTINSSON, *An $O(N)$ algorithm for constructing the solution operator to 2D elliptic boundary value problems in the absence of body loads*, Adv. Comput. Math., 40 (2014), pp. 773–796. (Cited on pp. 11, 58)
- [89] D. GORDON AND R. GORDON, *Robust and highly scalable parallel solution of the Helmholtz equation with large wave numbers*, J. Comput. Appl. Math., 237 (2013), pp. 182–196. (Cited on p. 13)
- [90] I. G. GRAHAM, E. A. SPENCE, AND E. VAINIKKO, *Recent results on domain decomposition preconditioning for the high-frequency Helmholtz equation using absorption*, in Modern Solvers for Helmholtz Problems, D. Lahaye, J. Tang, and K. Vuik, eds., Springer International, 2017, pp. 3–26. (Cited on pp. 12, 33)
- [91] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, SIAM, 1997, <https://doi.org/10.1137/1.9781611970937>. (Cited on p. 12)
- [92] M. N. GUDDATI AND J. L. TASSOULAS, *Continued-fraction absorbing boundary conditions for the wave equation*, J. Comput. Acoust., 8 (2000), pp. 139–156. (Cited on p. 10)

- [93] W. HACKBUSCH, *A sparse matrix arithmetic based on \mathcal{H} -matrices, Part I: Introduction to \mathcal{H} -matrices*, Computing, 62 (1999), pp. 89–108. (Cited on p. 11)
- [94] W. HACKBUSCH, *A sparse \mathcal{H} -matrix arithmetic, Part II: Application to multi-dimensional problems*, Computing, 64 (2000), pp. 21–47. (Cited on p. 11)
- [95] W. HACKBUSCH, *Hierarchical Matrices: Algorithms and Analysis*, Springer-Verlag, 2015. (Cited on p. 11)
- [96] T. HAGSTROM, *Radiation boundary conditions for the numerical simulation of waves*, Acta Numer., 8 (1999), pp. 47–106. (Cited on p. 8)
- [97] T. HAGSTROM AND S. I. HARIHARAN, *A formulation of asymptotic and exact boundary conditions using local operators*, Appl. Numer. Math., 27 (1998), pp. 403–416. (Cited on p. 10)
- [98] T. HAGSTROM, R. P. TEWARSON, AND A. JAZCILEVICH, *Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems*, Appl. Math. Lett., 1 (1988), pp. 299–302. (Cited on p. 13)
- [99] L. HALPERN, S. PETIT-BERGEZ, AND J. RAUCH, *The analysis of matched layers*, Confluentes Math., 3 (2011), pp. 159–236. (Cited on p. 8)
- [100] L. HALPERN AND L. N. TREFETHEN, *Wide-angle one-way wave equations*, J. Acoust. Soc. Amer., 84 (1988), pp. 1397–1404. (Cited on p. 38)
- [101] H. HAN AND X. WU, *Artificial Boundary Method*, Tsinghua University Press, Springer-Verlag, 2013. (Cited on p. 8)
- [102] J. HENRY, L. BENTO, AND M. SOARES, *Factorization of linear elliptic boundary value problems in non-cylindrical domains*, C. R. Acad. Sci. Paris Ser. I, 349 (2011), pp. 879–882. (Cited on pp. 39, 41)
- [103] K. L. HO AND L. GREENGARD, *A fast direct solver for structured linear systems by recursive skeletonization*, SIAM J. Sci. Comput., 34 (2012), pp. A2507–A2532, <https://doi.org/10.1137/120866683>. (Cited on p. 11)
- [104] A. J. HOFFMAN, M. S. MARTIN, AND D. J. ROSE, *Complexity bounds for regular finite difference and finite element grids*, SIAM J. Numer. Anal., 10 (1973), pp. 364–369, <https://doi.org/10.1137/0710033>. (Cited on p. 11)
- [105] Q. HU AND X. LI, *Efficient multilevel preconditioners for three-dimensional plane wave Helmholtz systems with large wave numbers*, Multiscale Model. Simul., 15 (2017), pp. 1242–1266, <https://doi.org/10.1137/16M1084791>. (Cited on p. 13)
- [106] F. IHLENBURG AND I. BABUŠKA, *Finite element solution of the Helmholtz equation with high wave number. Part I: The h -version of the FEM*, Comput. Math. Appl., 30 (1995), pp. 9–37. (Cited on p. 4)
- [107] K. ITO AND J. TOIVANEN, *Preconditioned iterative methods on sparse subspaces*, Appl. Math. Lett., 19 (2006), pp. 1191–1197. (Cited on p. 54)
- [108] B. N. KHOROMSKII AND G. WITTUM, *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface*, Springer-Verlag, 2004. (Cited on p. 19)
- [109] S. KIM AND M. LEE, *Artificial damping techniques for scalar waves in the frequency domain*, Comput. Math. Appl., 31 (1996), pp. 1–12. (Cited on p. 12)
- [110] S. KIM AND H. ZHANG, *Optimized Schwarz method with complete radiation transmission conditions for the Helmholtz equation in waveguides*, SIAM J. Numer. Anal., 53 (2015), pp. 1537–1558, <https://doi.org/10.1137/140980491>. (Cited on p. 13)
- [111] S. KIM AND H. ZHANG, *Optimized double sweep Schwarz method by complete radiation boundary conditions*, Comput. Math. Appl., 72 (2016), pp. 1573–1589. (Cited on pp. 14, 63)
- [112] R. KRESS, *Scattering by obstacles*, in Scattering: Scattering and Inverse Scattering in Pure and Applied Science, R. Pike and P. Sabatier, eds., Academic Press, 2002, pp. 191–210. (Cited on p. 50)
- [113] F. KWOK, *Optimized additive Schwarz with harmonic extension as a discretization of the continuous parallel Schwarz method*, SIAM J. Numer. Anal., 49 (2011), pp. 1289–1316, <https://doi.org/10.1137/090781632>. (Cited on p. 30)
- [114] R. F. LAGO, *A Study on Block Flexible Iterative Solvers with Applications to Earth Imaging Problem in Geophysics*, Ph.D. thesis, CERFACS, Toulouse, 2013. (Cited on p. 13)
- [115] P. LE TALLEC, *Domain decomposition methods in computational mechanics*, Comput. Mech. Adv., 1 (1994), pp. 121–220. (Cited on p. 13)
- [116] J. LI AND X. TU, *Convergence analysis of a balancing domain decomposition method for solving a class of indefinite linear systems*, Numer. Linear Algebra Appl., 16 (2009), pp. 745–773. (Cited on pp. 12, 13)
- [117] X. S. LI, *An overview of SuperLU: Algorithms, implementation, and user interface*, ACM Trans. Math. Softw., 31 (2005), pp. 302–325. (Cited on p. 11)

- [118] P.-L. LIONS, *On the Schwarz alternating method. I*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., SIAM, 1988, pp. 1–42. (Cited on p. 24)
- [119] F. LIU AND L. YING, *Additive sweeping preconditioner for the Helmholtz equation*, Multiscale Model. Simul., 14 (2016), pp. 799–822, <https://doi.org/10.1137/15M1017144>. (Cited on p. 14)
- [120] F. LIU AND L. YING, *Recursive sweeping preconditioner for the three-dimensional Helmholtz equation*, SIAM J. Sci. Comput., 38 (2016), pp. A814–A832, <https://doi.org/10.1137/15M1010154>. (Cited on p. 14)
- [121] F. LIU AND L. YING, *Sparsify and sweep: An efficient preconditioner for the Lippmann–Schwinger equation*, SIAM J. Sci. Comput., 40 (2018), pp. B379–B404, <https://doi.org/10.1137/17M1132057>. (Cited on p. 14)
- [122] H. LIU, *The Nine Chapters on the Mathematical Art: Companion and Commentary*, Oxford University Press, Science Press, Beijing, 1999 (originally written in 263 AD). (Cited on p. 11)
- [123] I. LIVSHITS, *The least squares AMG solver for the one-dimensional Helmholtz operator*, Comput. Vis. Sci., 14 (2011), pp. 17–25. (Cited on p. 13)
- [124] S. LOISEL, *Condition number estimates for the nonoverlapping optimized Schwarz method and the 2-Lagrange multiplier method for general domains and cross points*, SIAM J. Numer. Anal., 51 (2013), pp. 3062–3083, <https://doi.org/10.1137/100803316>. (Cited on pp. 13, 33)
- [125] Y. Y. LU, *Some techniques for computing wave propagation in optical waveguides*, Commun. Comput. Phys., 1 (2006), pp. 1056–1075. (Cited on pp. 9, 38)
- [126] P.-G. MARTINSSON AND V. ROKHLIN, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 205 (2005), pp. 1–23. (Cited on p. 11)
- [127] P.-G. MARTINSSON AND V. ROKHLIN, *A fast direct solver for scattering problems involving elongated structures*, J. Comput. Phys., 221 (2007), pp. 288–302. (Cited on p. 11)
- [128] L. C. MCINNES, R. F. SUSAN-RESIGA, D. E. KEYES, AND H. M. ATASSI, *Additive Schwarz methods with nonreflecting boundary conditions for the parallel computation of Helmholtz problems*, in Domain Decomposition Methods 10, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, 1998, pp. 325–333. (Cited on p. 13)
- [129] F. NATAF, *On the Use of Open Boundary Conditions in Block Gauss-Seidel Methods for Convection-Diffusion Equations*, Tech. Report, CMAP (École Polytechnique), 1993. (Cited on p. 24)
- [130] F. NATAF AND F. NIER, *Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains*, Numer. Math., 75 (1997), pp. 357–377. (Cited on pp. 10, 24)
- [131] F. NATAF AND F. ROGIER, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, Math. Models Methods Appl. Sci., 5 (1995), pp. 67–93. (Cited on pp. 61, 66)
- [132] F. NATAF, F. ROGIER, AND E. DE STUDER, *Domain decomposition methods for fluid dynamics*, in Navier-Stokes Equations and Related Nonlinear Analysis, A. Sequeira, ed., Plenum Press, 1995, pp. 367–376. (Cited on p. 33)
- [133] F. NATAF, F. ROGIER, AND E. DE STURLER, *Optimal Interface Conditions for Domain Decomposition Methods*, Tech. Report, CMAP (École Polytechnique), 1994. (Cited on pp. 7, 8, 13, 26, 61, 66)
- [134] J.-C. NÉDÉLEC, *Acoustic and Electromagnetic Equations: Integral Representations for Harmonic Problems*, Springer-Verlag, 2001. (Cited on pp. 4, 9, 10, 19, 22)
- [135] F. NIER, *Remarques sur les algorithmes de décomposition de domaines*, Sémin. Équ. Dériv. Partielles, École Polytech., Palaiseau, 1999, pp. 1–24. (Cited on p. 62)
- [136] L. N. OLSON AND J. B. SCHRODER, *Smoothed aggregation for Helmholtz problems*, Numer. Linear Algebra Appl., 17 (2010), pp. 361–386. (Cited on p. 13)
- [137] H. POINCARÉ, *La méthode de Neumann et le problème de Dirichlet*, Acta Math., 20 (1895), pp. 59–142. (Cited on p. 19)
- [138] J. POPOVIC AND O. RUNBORG, *Analysis of a fast method for solving the high frequency Helmholtz equation in one dimension*, BIT, 51 (2011), pp. 721–755. (Cited on p. 13)
- [139] J. POULSON, S. ENGQUIST, S. LI, AND L. YING, *A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations*, SIAM J. Sci. Comput., 35 (2013), pp. C194–C212, <https://doi.org/10.1137/120871985>. (Cited on pp. 14, 58, 63)
- [140] L. QIN AND X. XU, *On a parallel Robin-type nonoverlapping domain decomposition method*, SIAM J. Numer. Anal., 44 (2006), pp. 2539–2558, <https://doi.org/10.1137/05063790X>. (Cited on p. 13)

- [141] A. QUATERONI AND A. VALLI, *Domain Decomposition Methods for Partial Differential Equations*, Calderon Press, 1999. (Cited on pp. 13, 14, 24)
- [142] A. G. RAMM, *Fundamental solutions to some elliptic equations with discontinuous senior coefficients and an inequality for these solutions*, Math. Inequal. Appl., 1 (1998), pp. 99–104. (Cited on p. 22)
- [143] L. F. RICHARDSON, *The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam*, Philos. Trans. Roy. Soc. London Ser. A, 210 (1911), pp. 307–357. (Cited on p. 12)
- [144] C. D. RIYANTI, A. KONONOV, Y. A. ERLANGGA, C. VUIK, C. W. OOSTERLEE, R.-E. PLESSIX, AND W. A. MULDER, *A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation*, J. Comput. Phys., 224 (2007), pp. 431–448. (Cited on p. 12)
- [145] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, 2003, <https://doi.org/10.1137/1.9780898718003>. (Cited on p. 12)
- [146] M. SARKIS AND D. B. SZYLD, *Optimal left and right additive Schwarz preconditioning for minimal residual methods with Euclidean and energy norms*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1612–1621. (Cited on p. 54)
- [147] F. SAUVIGNY, *Partial Differential Equations 2*, 2nd ed., Springer-Verlag, 2012, pp. 248–257. (Cited on p. 22)
- [148] A. SCHÄDLE AND L. ZSCHIEDRICH, *Additive Schwarz method for scattering problems using the PML method at interfaces*, in Domain Decomposition Methods in Science and Engineering XVI, O. Widlund and D. E. Keyes, eds., Springer-Verlag, 2007, pp. 205–212. (Cited on pp. 13, 33, 49)
- [149] A. SCHÄDLE, L. ZSCHIEDRICH, S. BURGER, R. KLOSE, AND F. SCHMIDT, *Domain decomposition method for Maxwell’s equations: Scattering off periodic structures*, J. Comput. Phys., 226 (2007), pp. 477–493. (Cited on p. 33)
- [150] O. SCHENK, M. BOLLHÖFER, AND R. A. RÖMER, *On large-scale diagonalization techniques for the Anderson model of localization*, SIAM Rev., 50 (2008), pp. 91–112, <https://doi.org/10.1137/070707002>. (Cited on p. 11)
- [151] H. A. SCHWARZ, *Über einen Grenzübergang durch alternierendes Verfahren*, Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, 15 (1870), pp. 272–286. (Cited on p. 24)
- [152] A. H. SHEIKH, D. LAHAYE, AND C. VUIK, *On the convergence of shifted Laplace preconditioner combined with multilevel deflation*, Numer. Linear Algebra Appl., 20 (2013), pp. 645–662. (Cited on p. 13)
- [153] B. SMITH, P. E. BJORSTAD, AND W. D. GROPP, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996. (Cited on p. 13)
- [154] A. SOMMERFELD, *Partial Differential Equations in Physics*, Academic Press, 1949. (Cited on p. 9)
- [155] A. ST-CYR, M. J. GANDER, AND S. J. THOMAS, *Optimized multiplicative, additive, and restricted additive Schwarz preconditioning*, SIAM J. Sci. Comput., 29 (2007), pp. 2402–2425, <https://doi.org/10.1137/060652610>. (Cited on p. 28)
- [156] V. A. STEKLOV, *General Methods for the Solution of Principal Mathematical Physics Problems*, Kharkov Mathematical Society, 1901. (Cited on p. 19)
- [157] C. C. STOLK, *A rapidly converging domain decomposition method for the Helmholtz equation*, J. Comput. Phys., 241 (2013), pp. 240–252. (Cited on pp. 14, 31, 43, 49, 50, 51, 54, 63)
- [158] C. C. STOLK, *An improved sweeping domain decomposition preconditioner for the Helmholtz equation*, Adv. Comput. Math., 43 (2017), pp. 45–76. (Cited on pp. 14, 58)
- [159] C. C. STOLK, M. AHMED, AND S. K. BHOWMIK, *A multigrid method for the Helmholtz equation with optimized coarse grid corrections*, SIAM J. Sci. Comput., 36 (2014), pp. A2819–A2841, <https://doi.org/10.1137/13092349X>. (Cited on p. 13)
- [160] B. STUPFEL, *Improved transmission conditions for a one-dimensional domain decomposition method applied to the solution of the Helmholtz equation*, J. Comput. Phys., 229 (2010), pp. 851–874. (Cited on p. 13)
- [161] S. TA’ASAN, *Multigrid Methods for Highly Oscillatory Problems*, Ph.D. thesis, Weizmann Institute of Science, Rehovot, Israel, 1984. (Cited on p. 13)
- [162] A. TAFLOVE AND S. C. HAGNESS, *Computational Electrodynamics: The Finite Difference Time-Domain Method*, 3rd ed., Artech House, 2005. (Cited on pp. 41, 49)
- [163] A. TOSELLI, *Overlapping methods with perfectly matched layers for the solution of the Helmholtz equation*, in Eleventh International Conference on Domain Decomposition Methods, C. Lai, P. Bjorstad, M. Cross, and O. Widlund, eds., DDM.org, 1999, pp. 551–558. (Cited on pp. 13, 32)

- [164] A. TOSELLI AND O. WIDLUND, *Domain Decomposition Methods: Algorithms and Theory*, Springer-Verlag, 2005. (Cited on pp. 10, 13)
- [165] L. N. TREFETHEN, *The Definition of Numerical Analysis*, Tech. Report TR 92-1304, Cornell University, 1992. (Cited on p. 12)
- [166] L. N. TREFETHEN AND L. HALPERN, *Well-posedness of one-way wave equations and absorbing boundary conditions*, Math. Comp., 47 (1986), pp. 421–435. (Cited on p. 13)
- [167] M. B. VAN GIJZEN, Y. A. ERLANGGA, AND C. VUIK, *Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian*, SIAM J. Sci. Comput., 29 (2007), pp. 1942–1958, <https://doi.org/10.1137/060661491>. (Cited on p. 12)
- [168] P. VANEK, J. MANDEL, AND M. BREZINA, *Two-level algebraic multigrid for the Helmholtz problem*, in Domain Decomposition Methods 10, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, 1998, pp. 349–356. (Cited on p. 13)
- [169] A. VION AND C. GEUZAIN, *Double sweep preconditioner for optimized Schwarz methods applied to the Helmholtz problem*, J. Comput. Phys., 266 (2014), pp. 171–190. (Cited on p. 14)
- [170] S. WANG, M. V. DE HOOP, AND J. XIA, *Acoustic inverse scattering via Helmholtz operator factorization and optimization*, J. Comput. Phys., 229 (2010), pp. 8445–8462. (Cited on p. 11)
- [171] S. WANG, M. V. DE HOOP, AND J. XIA, *On 3D modeling of seismic wave propagation via a structured parallel multifrontal direct Helmholtz solver*, Geophys. Prospecting, 59 (2011), pp. 857–873. (Cited on pp. 11, 12)
- [172] S. WANG, X. S. LI, J. XIA, Y. SITU, AND M. V. DE HOOP, *Efficient scalable algorithms for solving dense linear systems with hierarchically semiseparable structures*, SIAM J. Sci. Comput., 35 (2013), pp. C519–C544, <https://doi.org/10.1137/110848062>. (Cited on p. 11)
- [173] C. WEISBECKER, P. AMESTOY, O. BOITEAU, R. BROSSIER, A. BUTTARI, J.-Y. L’EXCELLENT, S. OPERTO, AND J. VIRIEUX, *3D frequency-domain seismic modeling with a block low-rank algebraic multifrontal direct solver*, in SEG Houston 2013, 2013, pp. 3411–3416. (Cited on pp. 11, 12)
- [174] T. H. WOLFF, *Recent work on sharp estimates in second-order elliptic unique continuation problems*, J. Geometric Anal., 3 (1993), pp. 621–650. (Cited on p. 13)
- [175] J. XIA, *Efficient structured multifrontal factorization for general large sparse matrices*, SIAM J. Sci. Comput., 35 (2013), pp. A832–A860, <https://doi.org/10.1137/120867032>. (Cited on pp. 11, 12)
- [176] L. YING, *Sparsifying preconditioner for the Lippmann–Schwinger equation*, Multiscale Model. Simul., 13 (2015), pp. 644–660, <https://doi.org/10.1137/140985147>. (Cited on p. 14)
- [177] E. ZAUDERER, *Partial Differential Equations of Applied Mathematics*, 3rd ed., Pure Appl. Math. 71, John Wiley & Sons, 2011. (Cited on p. 50)
- [178] L. ZEPEDA-NÚÑEZ AND L. DEMANET, *Nested domain decomposition with polarized traces for the 2D Helmholtz equation*, SIAM J. Sci. Comput., 40 (2018), pp. B942–B981, <https://doi.org/10.1137/15M104582X>. (Cited on pp. 14, 59)
- [179] L. ZEPEDA-NÚÑEZ AND L. DEMANET, *The method of polarized traces for the 2D Helmholtz equation*, J. Comput. Phys., 308 (2016), pp. 347–388. (Cited on pp. 14, 43, 55, 58, 59, 63)
- [180] L. ZEPEDA-NÚÑEZ, R. J. HEWETT, AND L. DEMANET, *Preconditioning the 2D Helmholtz equation with polarized traces*, in SEG Technical Program Expanded Abstracts 2014, SEG, 2014, pp. 3465–3470. (Cited on pp. 14, 43, 55, 56, 58, 59)
- [181] L. ZEPEDA-NÚÑEZ AND H. ZHAO, *Fast alternating bidirectional preconditioner for the 2D high-frequency Lippmann–Schwinger equation*, SIAM J. Sci. Comput., 38 (2016), pp. B866–B888, <https://doi.org/10.1137/16M1064660>. (Cited on p. 14)
- [182] L. ZHU AND H. WU, *Preasymptotic error analysis of CIP-FEM and FEM for Helmholtz equation with high wave number. Part II: hp version*, SIAM J. Numer. Anal., 51 (2013), pp. 1828–1852, <https://doi.org/10.1137/120874643>. (Cited on p. 4)