

# Double sweep preconditioner for optimized Schwarz methods applied to the Helmholtz problem

A. Vion<sup>1</sup>, C. Geuzaine<sup>1,\*</sup>

*University of Liège, Department of Electrical Engineering and Computer Science,  
Montefiore Institute B28, B-4000 Liège, Belgium*

---

## Abstract

This paper presents a preconditioner for non-overlapping Schwarz methods applied to the Helmholtz problem. Starting from a simple analytic example, we show how such a preconditioner can be designed by approximating the inverse of the iteration operator for a layered partitioning of the domain. The preconditioner works by propagating information globally by concurrently sweeping in both directions over the subdomains, and can be interpreted as a coarse grid for the domain decomposition method. The resulting algorithm is shown to converge very fast, independently of the number of subdomains and frequency. The preconditioner has the advantage that, like the original Schwarz algorithm, it can be implemented as a matrix-free routine, with no additional preprocessing.

*Keywords:* Domain decomposition method, Helmholtz equation, preconditioners, iterative solvers, acoustic scattering, short-wave problem, finite element method

---

## 1. Introduction

Solving the Helmholtz equation numerically is a notoriously difficult problem, especially in the high-frequency regime, when the wavelength is much smaller than the size of the domain under study [1]. When solving the problem using a grid-based volume approach like the Finite Element Method (FEM), the number of unknowns becomes so large that the direct solution of the resulting linear system is computationally intractable. On the other hand, most iterative methods that have proved successful for elliptic problems become inefficient when applied to problems with highly oscillatory solutions [1].

Domain Decomposition Methods (DDM) try to combine both direct and iterative approaches, by decomposing the original domain into smaller subdomains over which a direct solution of the Helmholtz problem is possible, and then iterating over the subdomains until the solutions match. When the subdomains do not overlap, the problem can be reformulated in terms of unknown sources on the boundaries between the subdomains, linked to impedance-type transmission conditions between the subdomains. The solution produced by these sources inside the individual subdomains must match the restriction of the solution of the full problem on the subdomains. Such a formulation falls into the framework of Schwarz methods [2, 3, 4, 5]; recent improvements in the approximation of the transmission operators have made its convergence rate little sensitive to the wavenumber and discretization density [6, 7].

However, a remaining limitation of Schwarz methods is that, even with optimal transmission conditions, the number of iterations increases linearly with the number of subdomains. This limitation has been successfully addressed for certain classes of problems, by adding a component to the algorithm that is known in the DDM community as a “coarse grid” [8, 9, 10, 11, 12]. This generic name refers to any technique that enables global sharing of information between subdomains, while the basic additive algorithm only allows local

---

\*Corresponding author

*Email addresses:* a.vion@ulg.ac.be (A. Vion), cgeuzaine@ulg.ac.be (C. Geuzaine)

exchange of information, hence hampering convergence (note that multiplicative Schwarz algorithms enable long range exchange of information in one direction only, and cannot guarantee a convergence rate independent of the number of subdomains.) While coarse grids have proven to be very effective for Laplace-type problems, designing effective coarse grids for high-frequency Helmholtz problems proves difficult [11, 12].

Another way to look at the problem is to recast the DDM as the solution of a linear system, in which case it is natural to search for a preconditioner that would efficiently speed up the convergence of the solver. This paper explores that idea and contributes a way to precondition optimized Schwarz algorithms by taking advantage of recent advances in the development of efficient absorbing boundary conditions for the Helmholtz problem, used as transmission conditions. Starting from a propagation problem for which an exact expression of the optimal transmission condition exists, we will show how such a preconditioner can be designed by approximating the inverse of the iteration operator. It will be interpreted as a double sweep over the subdomains and works by propagating information, just as a coarse grid would. The resulting algorithm is shown to converge very fast, independently of the number of subdomains and frequency. This idea of sweeping to speed up the convergence of iterative Helmholtz solvers, yet not in the context of Schwarz methods, has been proposed in recent works [13, 14, 15] where it has shown a comparable effect on the rate of convergence. Our preconditioner has the advantage that, like the original algorithm, it can be implemented as a matrix-free routine and requires no additional preprocessing.

The paper is organized as follows. In Section 2 we begin by formulating the Schwarz algorithm as a linear problem amenable to a solution by Krylov subspace techniques. We then show in Section 3 that the iteration operator that corresponds to the problem formulated in terms of surface unknowns has different properties than the Helmholtz operator, that are especially interesting when a good approximation of the optimal transmission operator (the Dirichlet-to-Neuman map, or DtN map) is available. An iterative solution of this problem can be quickly obtained by means of an efficient preconditioner that exploits these properties, as detailed in Section 4. In Section 5, we summarize the different approximations of the DtN map that will be used as transmission conditions in the algorithm. Section 6 presents numerical results obtained with the proposed method on a variety of test cases.

## 2. Non-overlapping optimized Schwarz algorithm

The original domain decomposition method introduced by Schwarz [2] makes use of Dirichlet boundary conditions on the artificial interfaces. It is well known that the rate of convergence of this method depends on the size of the overlap between the subdomains and that the method stagnates if the subdomains do not overlap [8]. Convergence without overlap requires Robin or mixed conditions, or more generally impedance-type conditions [16, 17], giving rise to so-called non-overlapping optimized Schwarz algorithms [6, 5]. Such algorithms benefit from an easy partitioning of the domain and do not require the explicit construction of the normal derivative of the solution, although the treatment of junctions between multiple subdomains (so-called “cross-points”) requires special care [18].

In this paper, we take advantage of the structure of particular decompositions, called *layered partitionings*, in which the domain is sliced such that subdomains have at most two neighbours, and the first and last subdomains do not share a boundary (see Figure 1, right). We first give a detailed description of the basic algorithm, before rewriting it in a form suitable for Krylov acceleration.

### 2.1. Description of the algorithm

We want to solve the Helmholtz problem with wavenumber  $k$  in a domain  $\Omega$ :

$$\begin{aligned} -(\Delta + k^2)u &= f && \text{in } \Omega; \\ u &= u_D && \text{on } \Gamma_D; \\ u &\text{ is outgoing} && \text{on } \Gamma_S. \end{aligned} \tag{1}$$

We consider a layered decomposition of  $\Omega$  into  $N$  non-overlapping slices  $\Omega_i, 1 \leq i \leq N$ , with artificial boundaries  $\Sigma_{ij}$  between  $\Omega_i$  and  $\Omega_j$ , so that our partitioning contains no loop:  $\Omega_i \cap \Omega_j = \emptyset$  if  $|i - j| \neq 1$ .

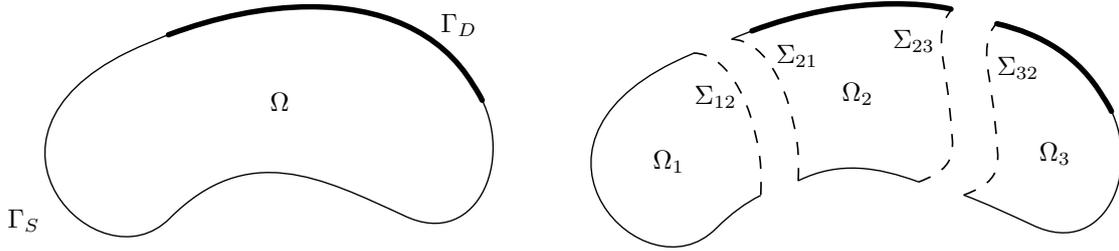


Figure 1: Left: A general computational domain  $\Omega$ . Absorbing boundary conditions (ABCs) are used on part of the boundary  $\Gamma_S$  to truncate the domain, and sources can be imposed via Dirichlet conditions on another part  $\Gamma_D$ . Right: Layered decomposition of  $\Omega$ . Artificial boundaries  $\Sigma_{ij}$  are introduced to separate subdomains  $\Omega_i$  and  $\Omega_j$ , such that each domain has 2 neighbours with the exception of the first and last domains.

The original problem (1) can be formulated in the subdomains so as to have  $u_i = u|_{\Omega_i}$ , by using impedance-matching boundary conditions on the artificial boundaries  $\Sigma_{ij}$ . Introducing the unknown interface data  $g = \{g_{ij}, 1 \leq i \neq j \leq N, |i - j| = 1\}$ , we look for the solution of:

$$\begin{aligned} -(\Delta + k^2)u_i &= 0 && \text{in } \Omega_i, \\ (\partial_n + \mathcal{S})u_i &= g_{ij} := (-\partial_n + \mathcal{S})u_j && \text{on } \Sigma_{ij}, \end{aligned} \quad (2)$$

where the operator  $\mathcal{S}$  has a twofold role: it must simulate the impedance of the domain that extends beyond the artificial boundary, and ensure that all sources located outside produce an equivalent contribution inside the subdomain. Boundary conditions on  $\partial\Omega_i \cap \partial\Omega$  are conserved from the original problem. For the sake of conciseness, we will abbreviate the definition of these problems as  $\mathcal{H}_i u_i = f_i$ . We will assume in all that follows that the DDM is well-posed, in the sense that each subproblem (2) is well-posed, i.e., away from interior resonances. We have defined a pair of unknown functions per interface, and we use the convention that  $g_{ij}$  is the impedance data for problem  $i$  on the common boundary with subdomain  $j$ . To solve for these new unknowns, the Schwarz algorithm works by iteratively solving the subproblems and transferring the updated information to the adjacent domains via an exchange relation. We present the additive version, where an iteration amounts to solving all subproblems in parallel (as opposed to the faster converging and sequential multiplicative version thereof):

$$\begin{aligned} -(\Delta + k^2)u_i^{(k+1)} &= 0 && \text{in } \Omega_i, \\ (\partial_n + \mathcal{S})u_i^{(k+1)} &= g_{ij}^{(k)} && \text{on } \Sigma_{ij}, \end{aligned} \quad (3)$$

and then to updating the unknowns:

$$\begin{aligned} g_{ij}^{(k+1)} &= -\partial_n u_j^{(k+1)} + \mathcal{S}u_j^{(k+1)} && \text{on } \Sigma_{ij}, \\ &= -g_{ji}^{(k)} + 2\mathcal{S}u_j^{(k+1)}. \end{aligned} \quad (4)$$

We still have the choice of operator  $\mathcal{S} : H^{1/2}(\Sigma_{ij}) \rightarrow H^{-1/2}(\Sigma_{ij})$ . (Strictly speaking, we should define 2 operators per artificial interface, but we use a unique notation for convenience.) We have the constraint that its null-space must be equal to  $\{0\}$ . It is well known that this choice influences the rate of convergence [5, 7], and it was shown in [19] that the optimum is obtained for  $\mathcal{S}$  being the exterior Dirichlet-to-Neumann (DtN) map  $\mathcal{D}$  of the complement of the subdomain  $\Omega \setminus \Omega_i$ , defined on a boundary  $\Sigma$  as:

$$\begin{aligned} \mathcal{D} : H^{1/2}(\Sigma) &\rightarrow H^{-1/2}(\Sigma), \\ u|_{\Sigma} &\mapsto \partial_n u|_{\Sigma} = \mathcal{D}u|_{\Sigma}. \end{aligned} \quad (5)$$

In practice, most optimized Schwarz algorithms make use of DtN maps that correspond to an open, free-space complement of the subdomain to avoid the very costly computation of the DtN map corresponding to

the actual complementary domain. This amounts to defining absorbing boundary conditions on the artificial interfaces. A perfectly matched layer (PML) can also be used for that purpose, as in [15]. Even in this case, using the exact DtN map is not practical for computing, as it is a non-local operator. We refer to Section 5 for the description of approximations that will prove useful for numerical applications.

Let us note that by defining a new set of unknowns  $g$ , we use the Schwarz procedure as a solver, although DDM algorithms are often regarded as preconditioners [8, 20]. In the latter case, the iterative solver acts on the full system obtained from the discretization of the original operator in the whole domain, with unknowns  $u$ .

## 2.2. Krylov acceleration

By exploiting the linearity of the problem, we can separate the (unknown) solutions of the subproblems into two components: the contribution of the artificial sources on the internal boundaries  $v_i$  and the physical sources  $w_i$ , such that  $u_i = v_i + w_i$ . In the course of iterations, we write the current approximation as  $u_i^{(k)} = v_i^{(k)} + w_i$ , since the physical sources do not vary. We then inject the decomposition of the unknown field in the update of the Schwarz unknowns (4):

$$\begin{aligned} g_{ij}^{(k+1)} &= -g_{ji}^{(k)} + 2\mathcal{S}v_j^{(k+1)} + 2\mathcal{S}w_j && \text{on } \Sigma_{ij}. \\ &= -g_{ji}^{(k)} + 2\mathcal{S}v_j^{(k+1)} + b_{ij} \end{aligned}$$

Considering the full vector of unknowns, we obtain the fixed point iteration:

$$g^{(k+1)} = \mathcal{A}g^{(k)} + b,$$

where the iteration operator  $\mathcal{A} : \times_{i,j=1}^N L^2(\Sigma_{ij}) \rightarrow \times_{i,j=1}^N L^2(\Sigma_{ij})$  is one step of the above algorithm with the physical sources set to 0. The vector  $b$  contains the local contributions of the external sources, and is computed as the output of the update relation (4) applied to  $w_i$ . From the iteration above we obtain, at convergence, the linear system:

$$\mathcal{F}g = (\mathcal{I} - \mathcal{A})g = b. \quad (6)$$

The new iteration operator  $\mathcal{F}$  is very similar to  $\mathcal{A}$ , with the difference that the update relation (4) is modified to match its definition:

$$g_{ij}^{(k+1)} = g_{ij}^{(k)} + g_{ji}^{(k)} - 2\mathcal{S}u_j^{(k+1)}. \quad (7)$$

The application of the iteration operator  $\mathcal{F}$  and the construction of the right-hand-side  $b$  are summarized in Algorithms 2.1 and 2.2. Once the Schwarz problem has been solved for  $g$ , one must compute  $v_i$  from (2) and finally obtain the original unknown  $u$  in each subdomain as  $u_i = v_i + w_i$ .

The Schwarz problem under form (6) can be interpreted as the solution of a linear system of equations, with matrix  $\mathcal{F}$ , unknowns  $g$  and right-hand-side  $b$ , ready to be solved by standard linear solvers suitable for non-symmetric systems. In particular, the solution can be accelerated by using a Krylov solver like GMRES [20, 21]. The construction of the Krylov subspace only requires matrix-vector products, which allows for a “matrix-free” implementation of the operator  $\mathcal{F}$ —the only operation required being the application of  $\mathcal{F}$  to a given vector, which amounts to solving the subproblems and performing the update of the unknowns. We will see in the next Section that an explicit expression of this matrix can actually be formed, although doing so is only useful for the purpose of analysis in view of the computational cost of the procedure.

## 3. Analysis of the iteration operator

As is well-known [19, 22], the above algorithm converges very slowly when applied to our layered decomposition with many subdomains, even with Krylov acceleration and the use of optimal transmission conditions on the artificial interfaces. Indeed, the reduction of the residual over the iterations exhibits large plateaus, whose length is proportional to the number of domains (see Figure 7). This behaviour is due to the fact that information can only be transferred from one subdomain to its neighbours in the course of

---

**Algorithm 2.1:** Application of the iteration operator  $g \leftarrow \mathcal{F}g$ .

---

```

// Solve subproblems (in parallel)
for  $i = 1 : N$ 
     $u_D \leftarrow 0$  on  $\partial\Omega_i \cap \Gamma_D$ 
     $f_i \leftarrow 0$ 
     $g_l \leftarrow g_{i,i-1}$ 
     $g_r \leftarrow g_{i,i+1}$ 
    Solve  $\mathcal{H}_i u_i = f_i$ , with impedance data  $g_l$  and  $g_r$ 
end

// Update impedance data
for  $i = 1 : N - 1$ 
     $g_{i,i+1} \leftarrow g_{i,i+1} + g_{i+1,i} - 2\mathcal{S}u_{i+1}|_{\Sigma_{i,i+1}}$ 
     $g_{i+1,i} \leftarrow g_{i+1,i} + g_{i,i+1} - 2\mathcal{S}u_i|_{\Sigma_{i+1,i}}$ 
end

```

---



---

**Algorithm 2.2:** Computation of the right-hand side  $b$ .

---

```

// Solve subproblems
for  $i = 1 : N$ 
     $u_D \leftarrow u_D$  on  $\partial\Omega_i \cap \Gamma_D$ 
     $f_i \leftarrow f$ 
     $g_l \leftarrow 0$ 
     $g_r \leftarrow 0$ 
    Solve  $\mathcal{H}_i u_i = f_i$ , with impedance data  $g_l$  and  $g_r$ 
end

for  $i = 1 : N - 1$ 
     $b_{i,i+1} \leftarrow -2\mathcal{S}u_{i+1}|_{\Sigma_{i,i+1}}$ 
     $b_{i+1,i} \leftarrow -2\mathcal{S}u_i|_{\Sigma_{i+1,i}}$ 
end

```

---

one iteration (local interactions). In this work we want to better understand the algebraic properties of the method in order to design a preconditioner that will improve these properties. We will see that a natural and successful solution is to enhance the information sharing among the subdomains by enabling long range communication between distant regions (global interactions).

We start by forming the explicit matrix  $F$  of the iteration operator  $\mathcal{F}$  for a layered decomposition in the 1D case; we will see in Section 3.3 that the extension to higher dimensions is straightforward in the context of matrix-free operators. At this stage, we have not precised yet how we are going to solve the PDEs that appear in algorithm (2). We compare two different approaches: first we define the “analytic” iteration operator  $\mathcal{F}_A$  when we use the analytical solution of the 1D Helmholtz equation in each subdomain as well as the exact expression of the exterior DtN map of an hypothetical complementary domain that would let the wave freely propagate without causing any reflection, leading to a perfectly absorbing (transparent, non-reflecting) boundary condition (in this particular case they are all identical:  $\mathcal{D} = -ik$ ). Then, in the more practical case where we use a numerical method and an imperfectly absorbing condition, we will observe the induced modifications to the structure of the associated “numerical” operator  $\mathcal{F}_N$ , mainly due to the induced spurious reflections at the artificial boundaries.



$$F_N(N) = \begin{bmatrix} \begin{array}{ccc|ccc} 1 & \epsilon & b_2 & & & \\ \epsilon & 1 & 0 & & & \\ \hline & 0 & 1 & \epsilon & & \ddots \\ & b_2 & \epsilon & 1 & & \\ \hline & & & & \ddots & \\ & & & & & b_{N-1} \\ \hline & & & & 0 & \\ & & & & & 1 & \epsilon \\ & & & & b_{N-1} & \epsilon & 1 \end{array} \end{bmatrix}. \quad (9)$$

The parameters  $b_i$  are also slightly affected. This is the consequence of the numerical dispersion (also known as the pollution effect) that arises upon discretization of the Helmholtz equation [26], as it causes inaccuracy of the absorbing boundary conditions. In other words, our choice of operator  $\mathcal{S} = -ik$  no longer matches the exact DtN map for the discretized problem, which causes a partial reflection of outgoing waves.

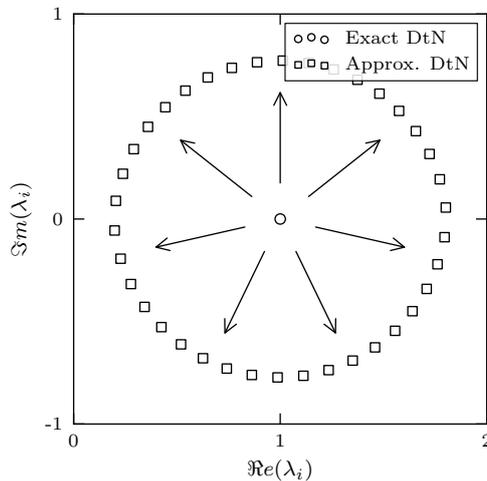


Figure 2: The eigenvalues of the iteration operator with exact DtN collapse to 1, as opposed to the case of imperfect impedance conditions where the eigenvalues tend to spread out over the complex plane, in a circle centered in  $(1, 0)$ . Despite the perfect conditioning of the system with the exact DtN, the algorithm does not converge faster, because the corresponding operator proves to be defective.

We observe that the eigenvalues are all distinct, with algebraic multiplicity 1, and tend to spread in the complex plane, in a circular pattern around  $(1, 0)$  (Figure 2). For 3 subdomains, they are:

$$\begin{aligned} \lambda_{1,2,3,4} &= 1 \pm \sqrt{\epsilon^2 \pm \epsilon b_2} \\ &\approx 1 \pm \sqrt{\pm \epsilon b_2} \end{aligned}$$

For more domains, the expression becomes too complicated to be reproduced here, yet they can still be seen as perturbations of 1:  $\lambda_i = 1 + r_i$ , where  $r_i$  is a complex number with approximately constant modulus  $|r_i| \approx R$  (if we suppose  $\epsilon \ll b_i$ ; small deviations appear for large  $N$ ) that corresponds to the radius of the circle. This radius tends to increase with the number of subdomains, or when a coarser discretization is used (further degrading the accuracy of the impedance condition). It asymptotically reaches 1 (with the consequence that some eigenvalues are close to 0), which strongly degrades the conditioning of the operator. Note that, as an effect of the cascaded square roots in the expression of the eigenvalues, even very small values of  $\epsilon$  have a strong effect on the spectral radius.



They are defined by:

$$\begin{aligned}\mathcal{B}_i^f &: H^{-1/2}(\Sigma_{i,i-1}) \rightarrow H^{-1/2}(\Sigma_{i,i+1}) \\ g_{i,i-1} &\longmapsto 2\mathcal{D}u_i(g_{i,i-1}, 0)|_{\Sigma_{i,i+1}} = \mathcal{B}_i^f g_{i,i-1}; \\ \mathcal{B}_i^b &: H^{-1/2}(\Sigma_{i,i+1}) \rightarrow H^{-1/2}(\Sigma_{i,i-1}) \\ g_{i,i+1} &\longmapsto 2\mathcal{D}u_i(0, g_{i,i+1})|_{\Sigma_{i,i-1}} = \mathcal{B}_i^b g_{i,i+1},\end{aligned}$$

where  $u_i(g_l, g_r)|_{\Sigma}$  refers to the restriction on boundary  $\Sigma$  of the solution to the subproblem  $\mathcal{H}_i u_i = f_i$  defined by (2), with the simplified notation  $g_l, g_r$  respectively corresponding to  $g_{i,i-1}, g_{i,i+1}$ .

These blocks are not necessarily square matrices, as the two artificial interfaces  $\Sigma_{i,i-1}$  and  $\Sigma_{i,i+1}$  that delimit a subdomain  $\Omega_i$  can be discretized with different number of unknowns; they are very expensive to form, but we can obtain their application to a vector by solving a problem and applying the (approximation of the) DtN map as in the definition above. We will see that this correspondence between the matrix blocks and the solution of subproblems will be very useful for the construction of a matrix-free version of our proposed preconditioner.

We mention for completeness that if approximate non-reflecting conditions are used or if internal reflections occur within the subdomains, additional blocks  $E_i^f$  and  $E_i^b$  appear in the matrix at the same positions as the  $\epsilon$  entries in the  $F_N$  matrix (9) of the 1D case. They correspond to operators  $\mathcal{E}_i^f$  and  $\mathcal{E}_i^b$  defined from an interface to itself:

$$\begin{aligned}\mathcal{E}_i^f &: H^{-1/2}(\Sigma_{i,i-1}) \rightarrow H^{-1/2}(\Sigma_{i,i-1}) \\ g_{i,i-1} &\longmapsto 2\mathcal{S}u_i(g_{i,i-1}, 0)|_{\Sigma_{i,i-1}} = \mathcal{E}_i^f g_{i,i-1}; \\ \mathcal{E}_i^b &: H^{-1/2}(\Sigma_{i,i+1}) \rightarrow H^{-1/2}(\Sigma_{i,i+1}) \\ g_{i,i+1} &\longmapsto 2\mathcal{S}u_i(0, g_{i,i+1})|_{\Sigma_{i,i+1}} = \mathcal{E}_i^b g_{i,i+1}.\end{aligned}$$

#### 4. Formal construction of the double sweep preconditioner

As said above, explicitly forming the matrix of the iteration operator is computationally expensive—and thus never done in practice except for analysis purposes. Assuming that a perfectly non-reflecting condition is used at the interfaces between the subdomains, and that no internal reflection occurs inside the subdomains, we will now show that the properties of the matrix can in fact be exploited to easily obtain its inverse: a recurrence relation exists that lets one build that inverse, even in the case of many subdomains, without forming the iteration matrix first. This process would provide the exact solution to the homogeneous 1D problem with exact DtN maps on the artificial and external boundaries, and can be used as a natural and efficient preconditioner in other cases.

##### 4.1. Inverse of the iteration operator with exact DtN

By exploiting the layered structure of the matrix  $F_A(N)$  obtained in Section 3.3, a recursion formula can be found for forming its inverse with increasing number of subdomains. Using operator notation, the

inverse has the following explicit form:

$$F_A^{-1}(N) = \begin{bmatrix} \begin{array}{ccc|ccc} I & & & -\mathcal{B}_2^b & & \\ & I & & & & \\ \hline & & & & I & \\ -\mathcal{B}_2^f & & & & & I \\ \hline \vdots & & & \vdots & & \ddots \\ \hline & & & & & I \\ \mathcal{F}_{M,2}^{-1} & \mathcal{F}_{M,4}^{-1} & \dots & -\mathcal{B}_{N-1}^f & & I \end{array} & \begin{array}{c} \mathcal{F}_{1,M-1}^{-1} \\ \mathcal{F}_{3,M-1}^{-1} \\ \vdots \\ -\mathcal{B}_{N-1}^b \\ I \end{array} \end{bmatrix}, \quad (11)$$

with the entries, using the index mappings defined in Appendix A:

$$\mathcal{F}_{mn}^{-1} = \begin{cases} -(-1)^{i(n)+j(m)} \prod_{k=i(n)}^{j(m)} \mathcal{B}_k^b & \text{if } m = 1, 3, \dots \text{ and } m < n; \\ -(-1)^{i(n)+j(m)} \prod_{k=j(m)}^{i(n)} \mathcal{B}_k^f & \text{if } m = 2, 4, \dots \text{ and } m > n; \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

The same operators  $\mathcal{B}_i$  appear in the iteration matrix and in its inverse, though the structure of the latter is more complex, with the operators multiplying each other.

One might think that forming and applying that inverse matrix will be expensive, but we will see in the next Section that the terms of the matrix-vector product with the inverse operator can be rewritten in such a way that the products of operators are fully avoided. This means that, provided that the  $\mathcal{B}_i$  operators are available, the matrix-vector products with the iteration operator and with its inverse can be obtained at reasonable cost and without having to form these matrices first. Since the application of  $\mathcal{B}_i$  to any vector  $v$  amounts to solving the  $i$ -th subproblem with  $v$  as impedance data on one side and applying operator  $\mathcal{S}$  to the restriction of the solution on the other side, we have all the ingredients at hand to build a fully matrix-free algorithm.

#### 4.2. Simplification and matrix-free version of the preconditioner

Throughout the developments above, we insisted on the recurrence relations in the structure of the iteration operator (10) and its approximate inverse (11), that we propose to use as a preconditioner. We now take advantage of that property to write the matrix-vector product with the preconditioner  $F_A^{-1}$  in a simplified way and give it an interpretation in terms of a double sweep of subproblems solves. The product  $g' = F_A^{-1}r$  (in the context of right-preconditioned Krylov solvers,  $r$  denotes the residual), is given component-wise as:

$$\begin{aligned} g'_{i,i-1} &= r_{i,i-1} + \sum_{p=i-1}^2 (-1)^{i+p} \prod_{q=i-1}^p \mathcal{B}_q^f r_{p,p-1}, \quad i = 2, \dots, N; \\ g'_{i,i+1} &= r_{i,i+1} + \sum_{p=i+1}^{N-1} (-1)^{i+p} \prod_{q=i+1}^p \mathcal{B}_q^b r_{p,p+1}, \quad i = 1, \dots, N-1. \end{aligned}$$

Factoring these expressions, we rewrite them as:

$$\begin{aligned} g'_{i,i-1} &= r_{i,i-1} - \mathcal{B}_{i-1}^f \left( r_{i-1,i-2} - \mathcal{B}_{i-2}^f \left( \dots \left( r_{3,2} - \mathcal{B}_2^f r_{2,1} \right) \right) \right), \\ &\quad i = 2, \dots, N; \\ g'_{i,i+1} &= r_{i,i+1} - \mathcal{B}_{i+1}^b \left( r_{i+1,i+2} - \mathcal{B}_{i+2}^b \left( \dots \left( r_{N-2,N-3} - \mathcal{B}_{N-1}^b r_{N-1,N} \right) \right) \right), \\ &\quad i = 1, \dots, N-1, \end{aligned}$$

which finally gives the double recurrence relation:

$$\begin{aligned}
g'_{21} &= r_{21}; \\
g'_{i+1,i} &= r_{i+1,i} - \mathcal{B}_i^f g'_{i,i-1}, \quad i = 2, \dots, N-1; \\
g'_{N-1,N} &= r_{N-1,N}; \\
g'_{i-1,i} &= r_{i-1,i} - \mathcal{B}_i^b g'_{i,i+1}, \quad i = N-1, \dots, 2.
\end{aligned}$$

The first relation describes the forward sequence: we start from the first boundary and propagate the information by solving a problem at each step to move to the next boundary and incorporate the contribution of that boundary. The other relation describes the same procedure in the backward direction; because these sequences are independent of each other, they can be done in parallel. We note that the extreme problems (the first and the  $N$ -th) are not solved in any of the sequences, so each of them requires  $N-2$  steps.

A sequence of solves over the domain is sometimes called a sweep, hence the name “double sweep” for this procedure. With the recurrence relations above, it becomes natural to implement the preconditioner as a matrix-free operator: each sweep is performed by calling, at every step, the subproblem solve routine and applying the DtN map to its output; the result is then used to start the next step. This gives rise to algorithm 4.1. In practice, the solution of  $\mathcal{H}_i u_i = f_i$  is done using the factorization of  $\mathcal{H}_i$  that is readily available in the Schwarz algorithm. But one could also envision using an approximate solution if a faster technique is available.

---

**Algorithm 4.1:** Application of the double sweep preconditioner  $r \leftarrow F_A^{-1} r$

---

<pre> // Forward sweep r<sub>21</sub> ← r<sub>21</sub> <b>for</b> i = 2 : N - 1     u<sub>D</sub> ← 0 on ∂Ω<sub>i</sub> ∩ Γ<sub>D</sub>     f<sub>i</sub> ← 0     g<sub>l</sub> ← r<sub>i,i-1</sub>     g<sub>r</sub> ← 0     Solve <math>\mathcal{H}_i u_i = f_i</math>, with data g<sub>l</sub> and g<sub>r</sub>     r<sub>i+1,i</sub> ← r<sub>i+1,i</sub> + 2S u<sub>i</sub> <sub>Σ<sub>i,i+1</sub></sub> <b>end</b> </pre>	<pre> // Backward sweep r<sub>N-1,N</sub> ← r<sub>N-1,N</sub> <b>for</b> i = N - 1 : 2     u<sub>D</sub> ← 0 on ∂Ω<sub>i</sub> ∩ Γ<sub>D</sub>     f<sub>i</sub> ← 0     g<sub>l</sub> ← 0     g<sub>r</sub> ← r<sub>i,i+1</sub>     Solve <math>\mathcal{H}_i u_i = f_i</math>, with data g<sub>l</sub> and g<sub>r</sub>     r<sub>i-1,i</sub> ← r<sub>i-1,i</sub> + 2S u<sub>i</sub> <sub>Σ<sub>i,i-1</sub></sub> <b>end</b> </pre>
---	--

---

One can further push the idea of the matrix representation of the iteration operator and the preconditioner, by directly considering the product of these matrices as our iteration operator instead of separately applying them. That matrix is computed as a classical matrix-matrix product, and can also be obtained by following a recurrence relation. This is somehow equivalent to defining a new, unpreconditioned algorithm, as it amounts to solve  $(F_N F_A^{-1}) g' = b$  with GMRES, with the matrix-vector product by  $(F_N F_A^{-1})$  performed in one single step. In the case of right preconditioning, that operation rewrites well in a matrix-free fashion as a double sequence of subproblems solutions, with the difference that it includes the first and last domains. The advantage of this combined approach is that it is more efficient in terms of number of solves per iteration, by skipping those solves that are redundant in the separate applications of the operators: the integrated version requires  $2N-2$  solves, for  $3N-4$  solves in the separate version. This results in Algorithm 4.2. (Note that in this case the solution of  $\mathcal{H}_i u_i = f_i$  must be computed fully.)

Computationally, although the sweeps in the algorithm are sequential by nature, parallelism can still be efficiently exploited in practice. Indeed, the most costly step (the factorization) of the direct solutions by sparse LU of all Helmholtz problems is fully parallelizable (the actual solution of each system at each iteration is also parallelizable by itself, but does not scale as well). It is also interesting to note that, on shared memory architectures, several sequential phases can be run concurrently for problems with multiple

excitations (e.g. multiple incidence angle or multiple source types). Finally, partial sweeps can also be envisioned on smaller groups of subdomains, which can restore some parallel efficiency [27].

---

**Algorithm 4.2:** Combined application of iteration operator and preconditioner  $r \leftarrow FF_A^{-1}r$

---

```

// Use two auxiliary variables:  $g^c$  contains the correction to the input data,  $g^t$ 
// saves data for use at next iteration. Both have same structure as input data  $r$ .

// Forward sweep
 $g_{2,1}^t \leftarrow 0$ 
for  $i = 2 : N$ 
     $g_l \leftarrow r_{i,i-1} + g_{i,i-1}^t$ 
     $g_r \leftarrow 0$ 
    Solve  $\mathcal{H}_i u_i = f_i$ 
     $g_{i-1,i}^c \leftarrow g_l - 2\mathcal{S}u_i|_{\Sigma_{i,i-1}}$ 
     $g_{i+1,i}^t \leftarrow 2\mathcal{S}u_i|_{\Sigma_{i,i+1}}$ 
end

// Backward sweep
 $g_{N-1,N}^t \leftarrow 0$ 
for  $i = N - 1 : 1$ 
     $g_l \leftarrow 0$ 
     $g_r \leftarrow r_{i,i+1} + g_{i,i+1}^t$ 
    Solve  $\mathcal{H}_i u_i = f_i$ 
     $g_{i+1,i}^c \leftarrow g_r - 2\mathcal{S}u_i|_{\Sigma_{i,i+1}}$ 
     $g_{i-1,i}^t \leftarrow 2\mathcal{S}u_i|_{\Sigma_{i,i-1}}$ 
end

// Add correction
 $r \leftarrow r + g^c$ 

```

---

#### 4.3. Spectrum of the preconditioned iteration operator

In Section 3, we have identified two sources for the slow convergence of the algorithm: the defectiveness of the operator, and the bad conditioning of the system caused by the small eigenvalues when many subdomains are used, in combination with approximate DtN maps on the artificial interfaces. Figure 4 shows the spectrum of the preconditioned iteration operator in the 1D case, i.e. of  $F_N F_A^{-1}$ . The eigenvalues of the preconditioned system are much more clustered around  $(1, 0)$  than in the unpreconditioned case. This spectrum resembles the one of the operator with the exact DtN, with the fundamental difference that the eigenvectors are now distinct. So the difficulty with the operator being defective, or close to it, has been removed and the good clustering of the eigenvalues opens the way for fast convergence. The preconditioner produces the same effect with a coarser discretization, yet the eigenvalues are not as well clustered in that case. In relation to this, we will see that the quality of the approximate DtN has a direct impact on the rate of convergence.

#### 4.4. Interpretation of the double sweep and related methods

We can interpret the application of the preconditioner as the transport of information over longer distances than the direct neighbourhood of the subdomains: the products of transmission operators  $\mathcal{B}_i$  that appear in (12) actually correspond to the contribution of a domain to another domain, that are separated by several other domains. This is illustrated on Figure 5.

We will see in Section 6 that, under the assumption that a good approximation of the DtN map is used, the double sweep preconditioner has the property that the convergence rate is independent of the number of subdomains. In the DDM community, a preconditioner that, at least partly, works by enabling global information sharing among the subdomains and thereby makes the convergence rate insensitive to the decomposition is known as a *coarse grid*. It was first introduced in [28] for the Laplace problem, and many authors have subsequently used that name, or sometimes *coarse space*, to describe a component of an algorithm or a technique (sometimes with no apparent connection with the original one) with similar properties [8, 9, 10, 11, 12]. Although our preconditioner is used to speed up the solution of a Schwarz problem, rather than the Helmholtz problem itself, we think that it makes sense in regards of its characteristics to view it as such a coarse grid.

The double sweep preconditioner is related to several other methods introduced over the last few years, not all of them linked to Domain Decomposition methods; in the realm of domain decomposition techniques,

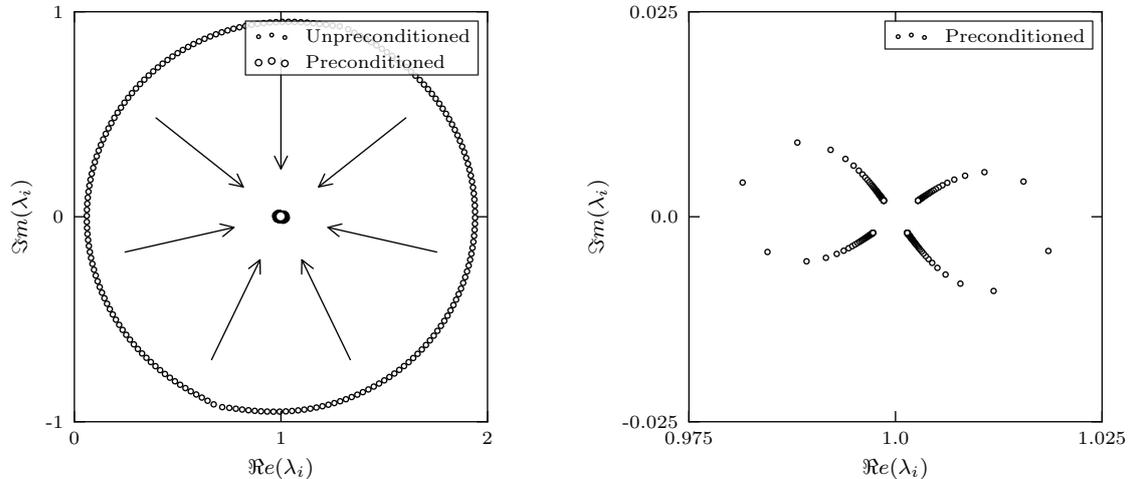


Figure 4: The double sweep preconditioner produces an excellent clustering of the eigenvalues of the iteration operator. Left: comparison of the spectrums of the operators ( $k = 40\pi$ ,  $N = 100$ ,  $n_\lambda = 20$ ), with and without preconditioner; right: zoom, centered in  $(1, 0)$ , on the eigenvalues of the preconditioned operator. The eigenvalues are again equal or very close to each other, but the preconditioned operator has a full set of distinct eigenvectors, so the convergence is now expected to be fast.

the sweeping algorithm was first analyzed by Nataf and Nier [22] for convection-diffusion operators, where the double-sweep was used as the actual solver, not as a preconditioner. Very recently, Stolk [15] proposed a domain decomposition approach to precondition the Helmholtz operator, with the same kind of layered decomposition as proposed in the present paper. The main difference is that we use the sweeps to precondition a Schwarz problem over interface unknowns, instead of preconditioning the original problem defined over the full set of volume unknowns. Also, the algorithm presented here is completely decoupled from the choice of the DtN map on the boundaries between the subdomains: as will be seen in the numerical tests (Section 6), in some cases even very inexpensive local approximations of the DtN can lead to rapid convergence of the preconditioned algorithm.

The idea of using sweeps to precondition the Helmholtz equation was also recently proposed by Engquist and Ying [13, 14] in a context unrelated to domain decomposition techniques. In this case preconditioners are built that are approximate factorizations of the discretized Helmholtz operator, obtained by eliminating the unknowns layer by layer in a sweeping process. When the size of the subdomains in our layered decomposition is very small (i.e., we keep a single layer of elements), the two methods present some similarities, although Engquist and Ying's sweeping PML preconditioner must be built in a preprocessing step. The parallel version of the sweeping PML preconditioner presented in [29] could be compared to our proposed double-sweep DDM preconditioner, with each subdomain factorization and subsequent backward solves made in parallel.

We close this section by mentioning the link between the double sweep preconditioner, the multiplicative Schwarz algorithm and a Gauss-Seidel iteration: they all share an intrinsically sequential behaviour, in contrast to additive Schwarz and Jacobi iterations, which are parallel by nature. Although the double sweep is not exactly a step of the multiplicative Schwarz algorithm, they are clearly linked and it is interesting to notice that an additive Schwarz method can be preconditioned by something that resembles the multiplicative version thereof.

## 5. Dirichlet-to-Neumann map

We mentioned earlier that the optimal choice of the transmission operator  $\mathcal{S}$  in the algorithm defined by (2) and (4) is the DtN map  $\mathcal{D}$  defined by (5). In that case the convergence of the iterative process is

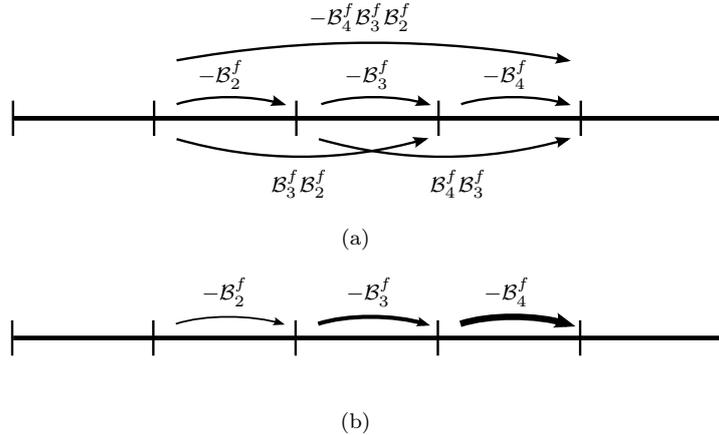


Figure 5: Coarse grid interpretation of the preconditioner in the 1D case: (a) the contribution of each artificial interface is transported to every other interfaces, via successive solves; (b) doing a sweep is equivalent, but the contributions of each domain are collected as one progresses in the sweep, as illustrated by the increasing stroke thickness of the arrows.

optimal [30, 9]. Beside the simple one-dimensional case where the DtN is trivial, this however leads to a very expensive procedure in practice, as the DtN is a non-local operator: see e.g. [3]. A great variety of techniques based on local transmission conditions have thus been proposed over the years: these include the class of FETI-H methods [31, 32, 10, 33], the optimized Schwarz approach [6], the evanescent modes damping algorithm [34, 35, 36] and the Padé-localized square-root operator [7]. All these local transmission conditions can be seen as approximations of the exact DtN operator; the better the related impedance operators approximate the exact DtN operator on all the modes of the solution, the better the convergence properties of the resulting DDM.

The DtN map is intimately linked with the artificial boundary conditions that were originally introduced to truncate infinite domains in various fields of application [37]. When used in the context of wave propagation problems, they are often called absorbing, non-reflecting or transparent boundary conditions. All these names refer to the fact that these conditions attempt to prevent or minimize the reflection of any outgoing wave that would hit the artificial boundary, therefore allowing it to freely leave the domain. This property is interesting when designing a domain decomposition algorithm since a wave propagating in a domain should naturally be able to cross artificial boundaries without distortion or reflection. An efficient transmission operator should therefore fully capture all of the outgoing waves and release them intact at the other side of the boundary.

We will use several local approximations of the DtN operator for the numerical tests presented in the next Section ( $\text{IBC}(\chi)$ ,  $\text{OO}_2$  and  $\text{GIBC}(N_p)$ ), as well as one non-local approximation using perfectly matched layers ( $\text{PML}(n_{\text{PML}})$ ).

### 5.1. Local impedance condition: $\text{IBC}(\chi)$

A simple local approximation of the DtN is the following Impedance Boundary Condition:

$$\mathcal{S}^{\text{IBC}(\chi)}u = (-ik + \chi)u,$$

where  $\chi$  is a self-adjoint positive operator [34]. When  $\chi = 0$ , one recovers the classical Sommerfeld radiation condition, used by Deprés in the original non-overlapping Schwarz method for Helmholtz [38]. We only consider here the case where  $\chi$  is a real-valued positive coefficient, which amounts to approximate the DtN map by a diagonal operator with a constant value on the diagonal. Choosing  $\chi \neq 0$  allows to improve the convergence of the DDM for evanescent modes. In Section 6.1 we will also use the impedance boundary condition with a modified wavenumber  $k_h$ , i.e.  $\mathcal{S}^{\text{IBC}_{k_h}(\chi)}u = (-ik_h + \chi)u$ .

### 5.2. Optimized second order local impedance condition: $OO_2$

While  $IBC(\chi)$  is a zero-th order approximation of the DtN, higher order approximations can also be constructed. For a generic transmitting boundary  $\Sigma$ , Gander et al. [6] proposed to construct a second order local approximation of the DtN in the form:

$$\mathcal{S}^{OO_2}u = (a - b\Delta_\Sigma)u, \quad (13)$$

where the complex numbers  $a$  and  $b$  are obtained by solving a min-max optimization problem on the rate of convergence (hence the name ‘‘Optimized Order 2’’), and  $\Delta_\Sigma$  is the Laplace-Beltrami operator on the interface  $\Sigma$ :  $\Delta_\Sigma := \text{div}_\Sigma \nabla_\Sigma$ . The optimal transmission coefficients depend on several parameters, chosen to solve the min-max problem on a bounded domain, excluding the Fourier modes close to the cut-off frequency [6]. In all the numerical tests we chose the parameters leading to the optimal convergence rate. (An optimized 0th order approximation can also be constructed in the same way.)

### 5.3. Padé-localized generalized impedance condition: $GIBC(N_p)$

Instead of a polynomial approximation of the DtN operator, a rational approximation was proposed in [7]:

$$\mathcal{S}^{GIBC(N_p)}u = C_0u + \sum_{\ell=1}^{N_p} A_\ell \text{div}_\Sigma (k_\varepsilon^{-2} \nabla_\Sigma) (1 + B_\ell \text{div}_\Sigma (k_\varepsilon^{-2} \nabla_\Sigma))^{-1}u, \quad (14)$$

where  $C_0$ ,  $A_\ell$  and  $B_\ell$  ( $\ell = 1, \dots, N_p$ ) are the coefficients of a complex Padé expansion of the square root operator (with a rotation of the branch cut of  $\pi/4$ ), and  $k_{\nu\epsilon} = k + \nu\epsilon$  is a complexified wavenumber. (In all subsequent tests we used  $\epsilon = k/4$ ). This condition leads to a DDM with quasi-optimal convergence properties, meaning that the rate of convergence is optimal on the evanescent modes and is improved compared to other local techniques for the remaining modes. To be noted is that, unlike the other techniques presented above,  $GIBC(N_p)$  makes use of  $N_p$  auxiliary unknown functions. Consequently, the size of the linear system to be solved is augmented. The additional cost is usually very small [7], unless the number of unknowns on the interfaces is large compared to number of volume unknowns.

### 5.4. Non-local PML condition: $PML(n_{PML})$

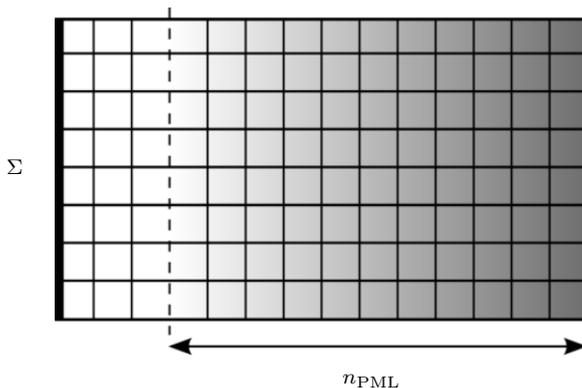


Figure 6: Geometry of fictitious ‘‘black-box’’ domain  $\Omega_{bb}$  used to compute the non-local approximation of the DtN map using a perfectly matched layer. The domain  $\Omega_{bb}$  is made of a few layers (3 in this case) between the interface of interest  $\Sigma$  and the PML, made of  $n_{PML}$  layers. The Dirichlet condition is imposed on  $\Sigma$ , while the boundary conditions on the top and bottom sides of  $\Omega_{bb}$  are inherited from the ones imposed on the edges of  $\Sigma$  in the original problem. The gradient represents the (1D) growth of the PML absorption factor.

Finally, we also consider a non-local approximation of the DtN operator constructed with the Perfectly Matched Layer (PML) technique. The procedure was developed in the context of matrix probing of the DtN map [39], that could be applied in our algorithm as suggested in [40]. A fictitious ‘‘black-box’’ domain  $\Omega_{bb}$

is first created by extruding the mesh of the interface  $\Sigma$  over  $n_{\text{PML}} + 3$  layers, using a mesh size identical to the mesh of the subdomain connected to  $\Sigma$  (see Figure 6). If the mesh of the interface  $\Sigma$  counts  $n_\Sigma$  internal nodes, the approximate  $n_\Sigma \times n_\Sigma$  DtN matrix is built by inspection: column  $c$  is obtained by solving a Helmholtz problem with an imposed value of 1 on the  $c^{\text{th}}$  node and 0 on the other nodes of  $\Sigma$ , and a 1D PML layer in the last  $n_{\text{PML}}$  cells of  $\Omega_{bb}$  (the boundary condition on the top and bottom boundary nodes of  $\Omega_{bb}$  is inherited from the ones on the top and bottom node of  $\Sigma$ ; a homogenous Dirichlet condition is imposed on the right). The normal derivative is extracted on  $\Sigma$  using the weak form of the finite element formulation. The number of layers (3 in all our numerical tests) added in front of the PML allow to clearly separate the PML from the elements used in the computation of the normal derivative.

This method would be too costly to use as-is in practical applications, but it is useful to consider it here for benchmarking the proposed preconditioner, as the quality of the approximate DtN map is directly related to that of the PML, which can be controlled via its thickness (number of layers  $n_{\text{PML}}$ ). We will thus refer to the non-local PML-based approximation as  $\text{PML}(n_{\text{PML}})$ . Another way of using PMLs in the algorithm would be to proceed like in [15]: instead of extracting the DtN map from the black-box as explained above, a PML is appended on the interfaces of the subdomains. Both approaches are algebraically equivalent, but their implementations differ in several aspects: with the black-box method, most of the computational work is done in preprocessing, and in the frequent practical cases where several interfaces are similar and can use the same DtN map (like in all the test cases of the next Section), the computation of the DtN map must be done only once; the other approach needs no additional preprocessing, but requires a modification of the geometry of all the subdomains and the evaluation of the derivative at the boundary of the PML, with an increased cost at every iteration. Recalling that a motivation for using a DDM is the size of the linear systems to be inverted, such an addition of the PMLs could make the method less attractive.

## 6. Numerical results

We first present numerical results in the 1D case with constant parameters, from which the preconditioner was derived in Section 4. We then use the same preconditioning strategy on more complex configurations and in the presence of a non-homogeneous medium, with guided and non-guided waves in 2D. In each case we analyse the effect of the accuracy of the DtN approximation on the performance of the preconditioner. In the non-homogeneous cases, we define the pulsation  $\omega$  and velocity  $c(x, y)$ , such that  $k(x, y) = \omega/c$ .

### 6.1. 1D with constant parameters

We study the behaviour of the algorithm in an homogeneous 1D medium  $\Omega = [0, 1]$ . This test case reproduces the conditions of the construction of the double sweep preconditioner (Section 4): we use absorbing conditions on both sides of the domain, and a volume source modeled by a delta-function located on the left boundary  $f = \delta(0)$ .

We observe that the convergence is fast compared to the unpreconditioned case (Figure 7): no more than a few iterations are required with the double sweep preconditioner. More interestingly, the rate of convergence is independent of the number of subdomains (Table 1) and wavenumber (Figure 8), under the condition that the exact DtN map is used. In 1D, this can be achieved by replacing the wavenumber  $k$  in the Sommerfeld condition  $\text{IBC}(0)$  by the wavenumber  $k_h$  accounting for the numerical dispersion of FEM for a discretization step  $h$ :

$$k_h = \frac{1}{h} \arccos \left( \frac{1 - \frac{(kh)^2}{3}}{1 + \frac{(kh)^2}{6}} \right).$$

In the more practical case when an approximate DtN map is used as transmission operator, the performance of the algorithm is slightly degraded and the number of iterations weakly depends on the number of subdomains. To observe this effect, we repeat the computation with different discretizations (we denote by  $n_\lambda$  the number of grid points per wavelength). As the numerical dispersion is more pronounced for coarser discretizations (smaller  $n_\lambda$ ) [41], the accuracy of the Sommerfeld conditions ( $\mathcal{S} = -ik$ ) that we use as transmission operators is also degraded.

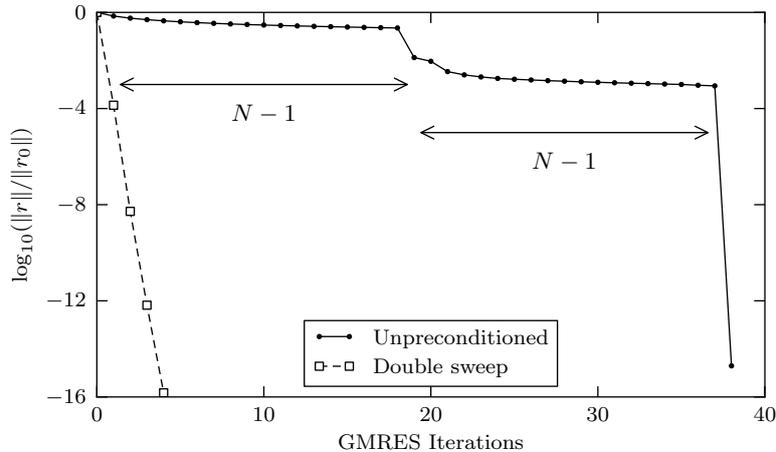


Figure 7: A typical convergence history of the optimized Schwarz algorithm with GMRES acceleration applied to a 1D problem, with  $N = 20$  subdomains (solid line). The curve exhibits 2 plateaus with length  $N - 1$ , before suddenly converging with full accuracy. The dashed line is obtained for the same problem with the preconditioned algorithm using optimal transmission operators.

	$N = 5$	25	50	100	150	200
IBC(0), $n_\lambda = 10$	4 (8)	4 (48)	5 (98)	5 (198)	6 (298)	6 (398)
IBC(0), $n_\lambda = 20$	3 (8)	3 (48)	4 (98)	4 (198)	4 (298)	4 (398)
IBC $_{k_h}$ (0), $n_\lambda = 10$	3 (8)	3 (48)	3 (98)	3 (198)	3 (298)	3 (398)
IBC $_{k_h}$ (0), $n_\lambda = 20$	2 (8)	2 (48)	2 (98)	2 (198)	2 (298)	3 (398)

Table 1: The iteration count ( $\|r\|/\|r_0\| \leq 10^{-6}$ ) in the 1D case is very small and steady with the number of domains when the exact DtN is used as transmission operator ( $\mathcal{S} = \mathcal{D} = -ik_h$ ); more iterations are required when less accurate approximations of the DtN map are used instead ( $\mathcal{S} \neq \mathcal{D} = -ik$ ), with a slight dependence on  $N$ . Values between parentheses are for the unpreconditioned algorithm:  $N_{it} = M = 2(N - 1)$ , where  $M$  is the size of the Schwarz system, hence the theoretical maximal number of iterations required to solve it by a Krylov method.

This behaviour is easily understood under the light of the analysis of Section 3: the double sweep preconditioner was built upon the assumption that no reflection occurs at the artificial boundaries, which can only be verified with an accurate DtN map as transmission operator; in that case, it is an exact inverse of the iteration operator and should converge in 1 iteration in exact arithmetic. With reflections at the artificial boundaries, the double sweep is no longer an exact inverse of the iteration operator and its performance deteriorates with the amplitude of the spurious reflected waves. More intuitively, an effect of the reflections is the transmission of partial information to the neighbouring subdomains. As this information is transmitted multiple times in the course of a sweep, it will be distorted everytime it crosses an artificial boundary. This explains why more iterations are required when more domains are involved if inexact DtN maps are used as transmission operators.

## 6.2. Homogeneous waveguide

The geometry of this test case is a straight waveguide ( $\Omega = [0, D] \times [0, d]$ ) made of an homogeneous medium. Homogeneous Dirichlet conditions are imposed on the upper and lower sides of the guides:  $u = 0$

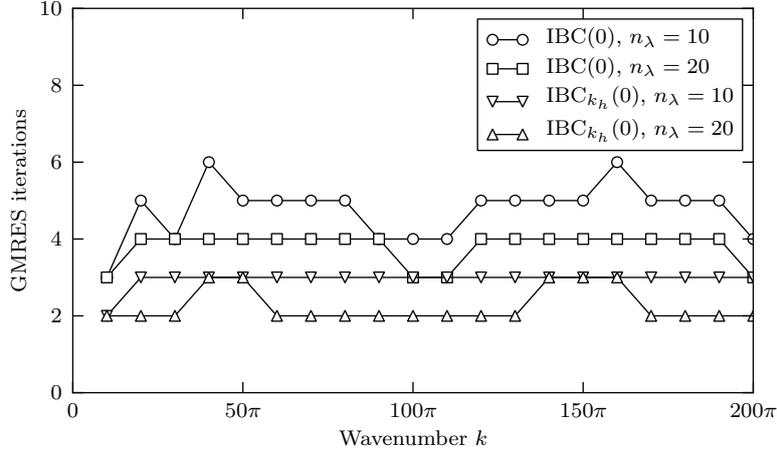


Figure 8: 1D problem: the iteration count is also stable with the wavenumber  $k$ , for all the tested transmission operators  $\mathcal{S}$  ( $N = 100$ ).

on  $y = \{0, d\}$ . We excite the second mode on the left hand side:  $u(0, y) = \sin(m\pi y/d)$ , with  $m = 2$ , and use an absorbing condition on the right hand side to model a continuing waveguide.

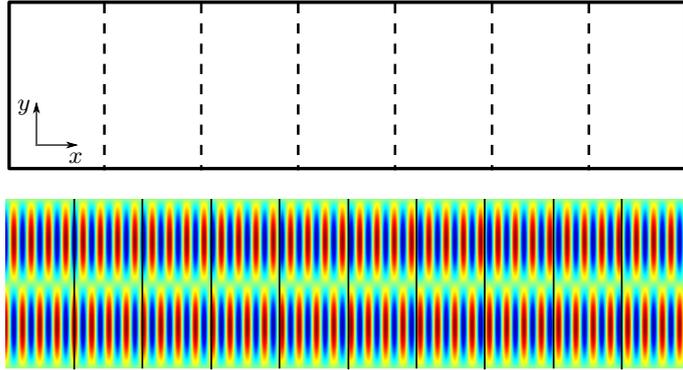


Figure 9: Waveguide geometry ( $D = 4; d = 1$ ) and decomposition, and solution for  $m = 2$  and  $k = 20\pi$ .

As there is only one possible propagation direction and no internal reflection for such a problem, the method is expected to behave somewhat like in the 1D case since we are still in the conditions of the preconditioner derivation. Moreover, an analytical solution for this problem is available for each mode, considered separately [42]:

$$u_A(x, y) = \sin\left(\frac{m\pi}{d}y\right)e^{i\beta_m x},$$

with:

$$\beta_m = \begin{cases} \sqrt{k^2 - (m\pi/d)^2} & \text{if } 1 \leq m \leq \frac{kd}{\pi} \quad (\text{propagative}); \\ i\sqrt{-k^2 + (m\pi/d)^2} & \text{otherwise} \quad (\text{evanescent}). \end{cases}$$

We only present results for the propagative modes. An exact expression of the DtN map can be inferred from the above expression:  $\mathcal{D} = -i\beta_m$ . Modifying the Sommerfeld transmission condition by replacing  $k$  by  $\beta_m$  therefore gives an excellent approximation of the DtN map, that can be ever improved by accounting for the numerical dispersion in its definition ( $\beta_m^h$ ). However, we do not include these results as they are not sufficiently general (only valid for a single mode.) The results for all the tested approximations of the DtN map are presented in Table 2. With the exception of  $\text{IBC}(\chi)$ , they all perform well and little penalty

	$\omega = 20\pi$					$\omega = 40\pi$				
	$N = 5$	10	25	50	100	5	10	25	50	100
IBC(0)	3 (8)	3 (18)	4 (48)	4 (98)	4 (198)	3	3	4	4	4
IBC( $k/2$ )	8 (8)	8 (18)	23 (50)	56 (120)	88 (326)	8	9	38	49	dnc
OO <sub>2</sub>	3 (8)	3 (18)	4 (46)	4 (98)	4 (201)	3	3	3	3	4
GIBC(2)	3 (8)	3 (18)	3 (48)	4 (119)	4 (239)	3	3	4	4	8
GIBC(8)	3 (8)	3 (18)	3 (48)	4 (119)	4 (240)	3	3	4	4	8
PML(5)	4 (8)	4 (18)	5 (48)	6 (96)	6 (196)	4	4	6	8	12
PML(15)	3 (8)	3 (18)	3 (48)	4 (98)	4 (198)	3	3	3	3	4
PML(75)	2 (8)	2 (18)	2 (48)	3 (98)	3 (198)	2	2	2	2	2

Table 2: Homogeneous waveguide: iteration count of the preconditioned GMRES ( $\|r\|/\|r_0\| < 10^{-6}$ ) as a function of  $N$ , for different transmission conditions. Values in parentheses are for the unpreconditioned algorithm; “dnc” stands for “did not converge” within the prescribed 500 iterations

is associated with the use of rough approximations of the DtN map; the good performance of the local approximations is therefore not surprising. GIBC( $k/2$ ) fails as it adds a real part to the operator  $\mathcal{S}$  while the DtN map in this particular case is purely imaginary.

### 6.3. Simple underground model

We consider a rectangular domain  $\Omega = [0, 600] \times [0, 1000]$  made of an heterogeneous medium with 3 different velocities in regions separated by straight non-parallel boundaries (Figure 10(a)). The outside world is modeled by Sommerfeld conditions on the “underground” sides, and a Neumann condition on the top side. A Dirichlet point source is located in the middle of the top line. This test-case was proposed in [43, 44] to test multigrid preconditioners for the Helmholtz equation.

As opposed to the previous test case, we are now in presence of abrupt variations of the wavenumber, that will produce internal reflections in different directions. The Sommerfeld conditions on the external boundaries are also likely to reflect part of the outgoing waves back into the domain. For these reasons, this problem is expected to be more challenging for a Schwarz method, as the multiple reflections will be harder to capture throughout the iterations. This also holds for our preconditioner, since this case is further removed from the hypotheses underlying its construction (recall that we neglected both numerical and physical reflections).

If the number of iterations is indeed slightly increased, the property of independence of the number of iterations with respect to the number of subdomains is actually preserved with the most accurate approximation of the DtN map (PML(75)). Even with less accurate approximations, we find that the number of additional iterations required for convergence grows relatively slowly with  $N$  (see Table 6.3). The GIBC(2) condition in particular performs extremely well—at a fraction of the cost of the non-local PML approximation.

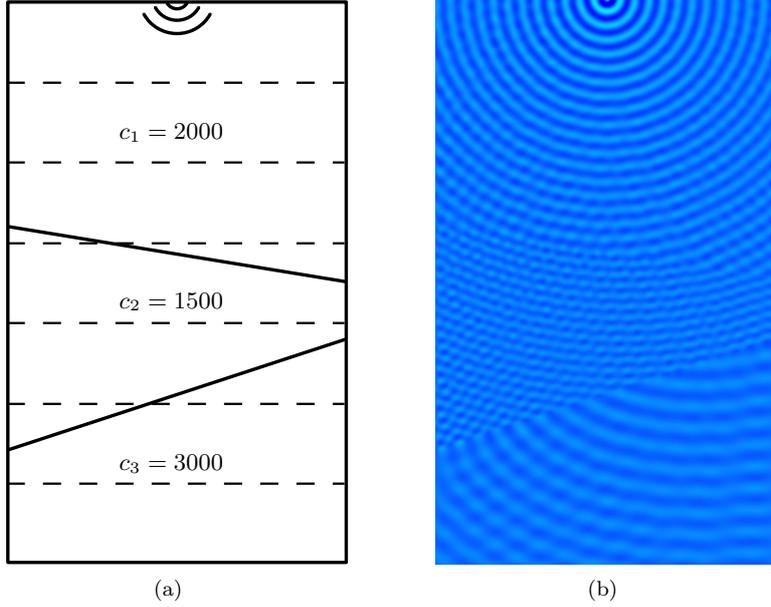


Figure 10: (a) Underground propagation test case geometry and decomposition, (b) solution for  $\omega = 160\pi$ .

	$\omega = 80\pi$					$\omega = 160\pi$				
	$N = 5$	10	25	50	100	5	10	25	50	100
IBC(0)	88 (96)	90 (133)	99 (244)	158 (415)	347 (dnc)	97	98	107	134	265
IBC( $k/4$ )	62 (70)	63 (110)	68 (231)	92 (404)	178 (dnc)	66	67	73	90	168
OO <sub>2</sub>	22 (38)	24 (77)	28 (207)	46 (384)	70 (dnc)	25	27	42	74	186
GIBC(2)	25 (40)	27 (74)	29 (186)	35 (369)	41 (dnc)	25	26	29	36	56
GIBC(8)	18 (36)	19 (73)	22 (186)	32 (369)	39 (dnc)	19	20	24	34	56
PML(5)	15 (38)	16 (75)	17 (195)	23 (368)	29 (dnc)	22	27	43	143	dnc
PML(15)	14 (36)	15 (74)	16 (183)	16 (359)	15 (dnc)	14	15	15	16	79
PML(75)	14 (35)	14 (72)	14 (182)	14 (357)	14 (dnc)	14	14	14	15	15

Table 3: Results for the underground propagation test case ( $\|r\|/\|r_0\| < 10^{-6}$ ) for different approximations of the DtN map. The convergence rate is independent of the number of domains and wavenumber when a very accurate (and computationally expensive) non-local approximation of the DtN map is used (PML(75)). Less accurate approximations degrade the convergence rate, especially when many subdomains are involved, but the algorithm still appears to be quite robust for the best fast local approximations; “dnc” stands for “did not converge” within the prescribed 500 iterations.

#### 6.4. Gaussian waveguide

The geometry of this test case is comparable to the one presented in Section 6.2, but we now consider a non-homogeneous medium, with a velocity profile constant in the propagation direction (along the  $x$ -axis) and gaussian in the transverse direction (see Figure 9):  $c(x, y) = 1.25(1 - .4e^{-32(y-.5)^2})$ . This a 2D version of a test case used in [29]. The second mode is excited on the left hand boundary and we model the continuing domain by means of a Sommerfeld condition on the right hand side.

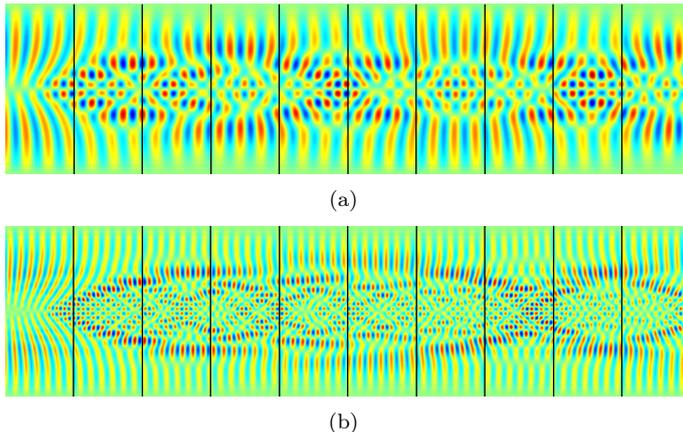


Figure 11: Solution of the gaussian waveguide problem at (a)  $\omega = 20\pi$  and (b)  $\omega = 40\pi$ . In both examples, the second mode is excited on the left hand side.

As opposed to the homogeneous case presented above, the gaussian speed profile will cause curved beams at high frequency that make the problem truly 2D, as can be seen on Figure 9. As a consequence, the “oscillations” exhibited in the matrix of the DtN map are more complex (their direction varies with position) and harder to capture by approximate methods. This is verified in practice: Table 4 shows that the convergence of all the local approximations degrades significantly for large  $N$ , and only the most accurate non-local PML approximations are able to maintain a constant iteration count with  $N$ . The interest of the preconditioner is particularly visible on this test case, where the unpreconditioned Schwarz algorithm becomes practically unusable for more than 25 domains.

## 7. Conclusion

In this paper we presented a double-sweep preconditioner for non-overlapping Schwarz methods applied to the Helmholtz problem. For a layered partitioning of the domain, the preconditioner works by propagating information by sweeping in both directions over the subdomains. The preconditioned algorithm can be implemented fully in a matrix-free fashion, requiring  $2N - 2$  subdomain solutions per iteration instead of  $N$  for the classical unpreconditioned Schwarz algorithm. The resulting algorithm, which can be interpreted as a coarse grid for the domain decomposition method, was shown to converge very fast, independently of the number of subdomains and frequency, provided that a good approximation of the DtN map on the interfaces between the subdomains is available.

Many perspectives exist for further investigation. First, since very accurate approximations of the DtN allow to obtain convergence in a constant number of iterations, one could investigate techniques that reduce the evaluation cost of accurate non-local DtN approximations, in particular the matrix probing technique [45] that takes advantage of a priori information on the structure of the DtN map [39]. Such a technique would shift most of the computational burden offline, in a preprocessing step. Second, by construction the algorithm would be particularly suitable for (quasi)-periodic problems, where most domains and DtN maps are identical. In such cases, the factorizations of the systems to be solved on each domain are identical and must be computed only once, as well as the DtN maps on their boundaries, leading to substantial savings

	$\omega = 20\pi$					$\omega = 40\pi$				
	$N = 5$	10	25	50	100	5	10	25	50	100
IBC(0)	38 (70)	40 (131)	78 (345)	207 (dnc)	497 (dnc)	51	65	129	217	dnc
IBC( $k/2$ )	35 (71)	45 (157)	134 (412)	314 (dnc)	dnc (dnc)	56	82	241	495	dnc
OO <sub>2</sub>	30 (62)	33 (128)	69 (356)	175 (dnc)	303 (dnc)	41	53	123	202	dnc
GIBC(2)	19 (53)	20 (114)	42 (314)	98 (dnc)	149 (dnc)	27	31	67	103	288
GIBC(8)	19 (53)	20 (114)	42 (314)	98 (dnc)	150 (dnc)	27	31	67	105	283
PML(5)	13 (47)	12 (103)	13 (271)	15 (dnc)	16 (dnc)	16	20	30	52	115
PML(15)	12 (44)	12 (101)	12 (266)	12 (dnc)	12 (dnc)	13	13	13	14	15
PML(75)	11 (44)	11 (99)	11 (264)	11 (dnc)	11 (dnc)	13	13	13	13	13

Table 4: Gaussian waveguide: iteration count of the GMRES ( $\|r\|/\|r_0\| < 10^{-6}$ ) as a function of  $N$ .

in the preprocessing step and memory usage. The homogeneous and gaussian waveguides considered as test cases in Section 6 are simple examples of such problems. Third, partial sweeps can also be envisioned on smaller groups of domains, which can restore some parallel efficiency [27]. Finally, the algorithm can be applied in a straightforward manner to other harmonic wave-type problems, like the propagation of electromagnetic waves [27].

### Acknowledgement

This work was supported in part by the Belgian Science Policy (PAI grant P7/02) and the Walloon Region (WIST3 grants ONELAB and ALIZEES).

The authors are very grateful to Prof. L. Demanet and R. Bélanger-Rioux from MIT for fruitful discussions about this work.

## Appendix A. Matrix form of the iteration operator

Here we detail how the matrix of the iteration operator  $F_A$  (8) was formed in Section 3.1, when analytical solutions of the PDEs in the Schwarz algorithm are used. Applying algorithm (3–4) with Krylov acceleration, we have to solve the subproblems (recall that external sources are cancelled in the definition of operator  $\mathcal{F}$  and that we use  $\mathcal{S} = \mathcal{D} = -ik$ , so we apply the same condition on the artificial interfaces than on the external boundaries):

$$\begin{aligned} -(\partial_{xx} + k^2)u_j &= 0 && \text{in } \Omega_j \\ (\partial_n - ik)u_j &= g_{ji} && \text{on } \Sigma_{ij} \\ (\partial_n - ik)u_j &= 0 && \text{on } \{x_l, x_r\}. \end{aligned} \quad (\text{A.1})$$

The solutions of these problems are a superposition of a forward and a backward wave:

$$u_j(x, g_l, g_r) = A_{f,j}e^{ikx} + A_{b,j}e^{-ikx}, \quad x \in \Omega_j, \quad j = 1, \dots, N,$$

with respective amplitudes given as functions of the impedance data  $g_l$  and  $g_r$  on the artificial boundary (we suffix by  $l$  and  $r$  quantities or coordinates associated with resp. the left and right side of the considered domain):

$$A_{f,j} = -\frac{g_l}{2ik}e^{ikx_l} \quad ; \quad A_{b,j} = -\frac{g_r}{2ik}e^{ikx_r}, \quad (\text{A.2})$$

While the matrix is easy to form numerically, its formal expression is rather technical and requires some definitions. We first number the unknowns of the Schwarz algorithm as  $g = [g_{12}, g_{21}, g_{23}, \dots]^T$ , where an unknown  $g_{ij}$  corresponds to the impedance data of the boundary condition for problem  $i$ , on  $\Sigma_{ij}$ . There are two unknowns per artificial interface, for a total of  $M = 2(N - 1)$  unknowns. Then, we classically number the entries of the matrix as:

$$F = \begin{bmatrix} F_{11} & \dots & F_{1M} \\ \vdots & \ddots & \vdots \\ F_{M1} & \dots & F_{MM} \end{bmatrix},$$

where entry  $F_{mn}$  refers to the  $m$ -th row and  $n$ -th column. So we have two different ways of indexing the unknowns: the ‘‘matrix index’’  $g_m$  and the ‘‘problem index’’  $g_{ij}$ . As both are convenient depending on the context, we define the index mappings:

$$\begin{aligned} m(i, j) &= \begin{cases} i + j - 1 & \text{if } i > j \\ i + j - 2 & \text{if } i < j \end{cases} \quad ; \\ [i, j](m) &= \begin{cases} [\frac{m+1}{2}, \frac{m+3}{2}] & \text{if } m = 1, 3, \dots \\ [\frac{m}{2} + 1, \frac{m}{2}] & \text{if } m = 2, 4, \dots \end{cases} \end{aligned}$$

Each unknown  $g_{ij}$  has a companion unknown  $g_{ji}$  associated to the same interface  $\Sigma_{ij}$ . In the matrix indexing, we will write  $g'_m$  the unknown associated to  $g_m$ , with the index relation:

$$m' = m - (-1)^m.$$

The column  $F_{\cdot,n}$  is the output of the update relation (7) applied to the  $n$ -th column of an identity matrix as source vector  $g^k$ , so the  $m$ -th entry of that column will be:

$$F_{mn} = \delta_{mn} + \delta_{m'n} + 2iku_j(m)(x_{\Sigma_{ij}(m)}, \delta_{2(j(m)-1),n}, \delta_{2j(m)-1,n}). \quad (\text{A.3})$$

In the  $n$ -th column, at most 3 entries have a non-zero contribution from at least one of its terms: the  $n$ -th entry (the one on the diagonal of the matrix) is exactly 1, as only the  $\delta_{nn}$  term contributes; the  $n'$ -th is 0,

because the  $\delta_{m'n}$  term is exactly canceled by the problem contribution as an effect of the exact transmission condition; and the  $(n\pm 2)$ -th only has the problem contribution alone. The matrix therefore has the following structure for  $N$  subdomains:

$$F_A(N) = \begin{bmatrix} \begin{array}{cc|c} 1 & 0^* & b_2 \\ 0^* & 1 & 0 \\ \hline 0 & 1 & 0^* \\ b_2 & 0^* & 1 \end{array} & & \\ & \ddots & \\ & & b_{N-1} \\ & & 0 \\ & & 1 & 0^* \\ & & b_{N-1} & 0^* & 1 \end{bmatrix},$$

where parameters  $b_i$  depend only on the size of the subdomains  $\Delta_i = x_{\Sigma_{i+1}} - x_{\Sigma_i}$  (we introduce the simplified notation  $\Sigma_k = \Sigma_{ij}$ , with  $k = \min(i, j)$ ):  $b_i = e^{ik\Delta_i}$ . The  $0^*$  entries indicate values that are 0 as the result of the cancellation of 2 contributions in expression (A.3) (we will see that in the numerical solution case, they are no longer 0.) The condition number of  $F_A(N)$  is small. For example, for the case of 3 subdomains, the matrix is:

$$F_A(3) = \begin{bmatrix} \begin{array}{cc|c} 1 & 0^* & b_2 \\ 0^* & 1 & 0 \\ \hline 0 & 1 & 0^* \\ b_2 & 0^* & 1 \end{array} \end{bmatrix}.$$

One can easily verify that all eigenvalues are equal to 1 ( $\lambda_{1-4} = 1$ ) and that only 2 independent eigenvectors exist:  $[1, 0, 0, 0]$  and  $[0, 0, 0, 1]$ . They are indeed the only possible invariants since the wave they produce leaves the domain via the external boundaries instead of coupling to other subdomains. Since  $b_2$  has unit modulus ( $|b_{i, 1 < i < N}| = 1$ ), we have for the condition number, in spectral norm:

$$\|F_A(3)\|_2 = \|F_A^{-1}(3)\|_2 = \sqrt{\frac{1}{2}|b_2|^2 + \frac{1}{2}\sqrt{|b_2|^4 + 4|b_2|^2} + 1};$$

$$\kappa_2(F_A(3)) = \|F_A(3)\|_2 \|F_A^{-1}(3)\|_2 = \frac{|b_2|^2}{2} + \frac{\sqrt{|b_2|^4 + 4|b_2|^2}}{2} + 1 = \left(\frac{|b_2|}{2} + \sqrt{\frac{|b_2|^2}{4} + 1}\right)^2 = 2.618.$$

The condition number grows linearly with the number of subdomains, as shown by Table A.5.

$N$	2	3	4	5	10	20	50	100
$\kappa_2$	1	2.62	4.09	5.41	11.94	24.57	63	126.67

Table A.5: Condition number of the iteration matrix  $F_A(N)$  for increasing number of subdomains.

## References

- [1] O. Ernst, M. Gander, Why it is difficult to solve Helmholtz problems with classical iterative methods, Numerical Analysis of Multiscale Problems.
- [2] H. A. Schwarz, Über einen grenzübergang durch alternierendes verfahren, Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich 15 (1870) 272–286.
- [3] F. Collino, S. Ghanemi, P. Joly, Domain decomposition method for harmonic wave propagation: a general presentation, Comput. Methods Appl. Mech. Engrg. 184 (2–4) (2000) 171–211.
- [4] A. Toselli, O. Widlund, Domain decomposition methods – algorithms and theory, Ser. Comput. Math., Springer, 2005.
- [5] M. J. Gander, Optimized Schwarz methods, SIAM J. Numer. Anal. 44 (2) (2006) 699–731.

- [6] M. Gander, F. Magoulès, F. Nataf, Optimized Schwarz methods without overlap for the Helmholtz equation, *SIAM Journal of Scientific Computing* 24 (1) (2002) 38–60.
- [7] Y. Boubendir, X. Antoine, C. Geuzaine, A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation, *Journal of Computational Physics* 231 (2).
- [8] B. F. Smith, P. E. Bjorstad, W. D. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
- [9] F. X. Roux, F. Magoulès, L. Series, Y. Boubendir, Approximation of optimal interface boundary conditions for two-Lagrange multiplier FETI method, *Lecture Notes in Computational Science and Engineering* 40 (2005) 283–290.
- [10] C. Farhat, R. Tezaur, J. Toivanen, A domain decomposition method for discontinuous Galerkin discretizations of Helmholtz problems with plane waves and lagrange multipliers, *International Journal for Numerical Methods in Engineering* 78 (2009) 1513–1531.
- [11] O. Dubois, M. J. Gander, S. Loisel, A. St-cyr, D. B. Szyld, The optimized Schwarz method with a coarse grid correction, *SIAM J. Scientific Computing* 34 (2).
- [12] M. Gander, H. Zhang, Domain decomposition methods for the helmholtz equation: A numerical investigation, in: R. Bank, M. Holst, O. Widlund, J. Xu (Eds.), *Domain Decomposition Methods in Science and Engineering XX*, Vol. 91 of *Lecture Notes in Computational Science and Engineering*, Springer Berlin Heidelberg, 2013, pp. 215–222.
- [13] B. Engquist, L. Ying, Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers, *Multiscale Model. Simul.* 9 (2) (2011) 686–710.
- [14] B. Engquist, L. Ying, Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation, *Communications on Pure and Applied Mathematics* 64 (5) (2011) 697–735.
- [15] C. C. Stolk, A rapidly converging domain decomposition method for the Helmholtz equation, *Journal of Computational Physics* 241 (0) (2013) 240 – 252.
- [16] P.-L. Lions, On the Schwarz alternating method III: a variant for non overlapping subdomains, in: T. Chan, R. Glowinski, J. Périaux, O. Widlund (Eds.), *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, held in Houston, Texas, March 20–22, 1989, SIAM, Philadelphia, PA, 1990, pp. 202–223.
- [17] Després, Décomposition de domaine pour les problèmes de propagation d’ondes en régime harmonique. le théorème de Borg pour l’équation de Hill vectorielle, Ph.D. thesis, Paris VI University, France (1991).
- [18] Y. Boubendir, A. Bendali, Dealing with cross-points in a non-overlapping domain decomposition solution of the Helmholtz equation, in: *Mathematical and Numerical Aspects of Wave Propagation WAVES 2003*, Springer Berlin Heidelberg, 2003, pp. 319–324.
- [19] F. Nataf, Interface connections in domain decomposition methods, in: *Modern methods in scientific computing and applications (Montréal, QC, 2001)*, Vol. 75 of *NATO Sci. Ser. II Math. Phys. Chem.*, Kluwer Acad. Publ., Dordrecht, 2002, pp. 323–364.
- [20] A. Greenbaum, *Iterative methods for solving linear systems*, Vol. 17 of *Frontiers in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.
- [21] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd Edition, Society for Industrial and Applied Mathematics, 2003.
- [22] F. Nataf, F. Nier, Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains, *Numer. Math.* 75 (1997) 357–377.
- [23] G. Strang, *Linear algebra and its applications*, 3rd Edition, Thomson Brooks/Cole learning, 1988.
- [24] N. Chen, Inverse iteration on defective matrices, *Mathematics of Computation* 31 (139) (1977) 726–732.
- [25] J. Zhongxiao, The convergence of Krylov subspace methods for large unsymmetric linear systems, *Acta Mathematica Sinica* 14 (4) (1998) 507–518.
- [26] F. Ihlenburg, I. Babuska, Finite element solution to the Helmholtz equation with high wavenumber. part I: The h-version of the FEM, *Comput. Math. Appl.* 39 (1995) 9–37.
- [27] A. Vion, C. Geuzaine, Parallel double sweep preconditioner for the optimized Schwarz algorithm applied to high frequency Helmholtz and Maxwell equations, in: *Submitted to the Proceedings of the 22th International Conference on Domain Decomposition Methods (DD22)*, Lugano, Italy, 2013.
- [28] M. Dryja, O. Widlund, Some domain decomposition algorithms for elliptic problems, in: L. Hayes, D. Kincaid (Eds.), *Iterative Methods for Large Linear Systems*, Academic Press, San Diego, CA, 1989, pp. 273–291.
- [29] J. Poulson, B. Engquist, S. Li, L. Ying, A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations, *SIAM Journal on Scientific Computing*, under review.
- [30] F.-X. Roux, F. Magoulès, S. Salmon, L. Series, Optimization of interface operator based on algebraic approach, In *Domain Decomposition Methods in Sci. Engrg.* (2002) 297–304.
- [31] C. Farhat, A. Macedo, M. Lesoinne, A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems, *Numerisch Mathematik* 85 (2) (2000) 282–303.
- [32] C. Farhat, A. Macedo, M. Lesoinne, F. Roux, F. Magoulès, A. D. L. Bourdonnaye, Two-level domain decomposition methods with lagrange multipliers for the fast iterative solution of acoustic scattering problems, *Computer Methods in Applied Mechanics and Engineering* 184 (2000) 213–239.
- [33] A. D. L. Bourdonnaye, C. Farhat, A. Macedo, F. Magoulès, F. Roux, A non overlapping domain decomposition method for the exterior Helmholtz problem, *Contemporary Mathematics* 218 (2) (1998) 42–66.
- [34] Y. Boubendir, An analysis of the BEM-FEM non-overlapping domain decomposition method for a scattering problem, *Journal of Computational and Applied Mathematics* 204 (2) (2007) 282–291.
- [35] Y. Boubendir, A. Bendali, M. B. Fares, Coupling of a non-overlapping domain decomposition method for a nodal finite element method with a boundary element method, *International Journal for Numerical Methods in Engineering* 73 (2008) 1624–1650.

- [36] A. Bendali, Y. Boubendir, M. Fares, A FETI-like domain decomposition method for coupling FEM and BEM in large-size problems of acoustic scattering, *Computer & Structures* 85 (2007) 526–535.
- [37] D. Givoli, Numerical methods for problems in infinite domains, Vol. 33 of *Studies in Applied Mechanics*, Elsevier Scientific Publishing Co., Amsterdam, 1992.
- [38] B. Després, Domain decomposition method and the Helmholtz problem, in: G. Cohen, L. Halpern, P. Joly (Eds.), *Mathematical and numerical aspects of wave propagation phenomena*, SIAM, Philadelphia, PA, 1991, pp. 44–52.
- [39] R. Bélanger-Rioux, L. Demanet, Compressed absorbing boundary conditions via matrix probing, *ArXiv e-prints*, arXiv:1401.4421.
- [40] A. Vion, R. Bélanger-Rioux, L. Demanet, C. Geuzaine, A DDM double sweep preconditioner for the Helmholtz equation with matrix probing of the DtN map, in: *Mathematical and Numerical Aspects of Wave Propagation WAVES 2013*, 2013.
- [41] F. Ihlenburg, I. Babuska, Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation, *International Journal for Numerical Methods in Engineering* 38 (1995) 3745–3774.
- [42] B. Pinçon, K. Ramdani, Selective focusing on small scatterers in acoustic waveguides using time reversal mirrors, *Inverse Problems* 23 (1) (2007) 1–25.
- [43] R. Plessix, W. Mulder, Separation-of-variables as a preconditioner for an iterative Helmholtz solver, *Applied Numerical Mathematics* 44 (2003) 385–400.
- [44] Y. A. Erlangga, C. Vuil, C. W. Oosterlee, Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation, *Applied Numerical Mathematics* 56 (2006) 648–666.
- [45] J. Chiu, L. Demanet, Matrix probing and its conditioning, *SIAM J. Numer. Anal.* 50 (2012) 171–193.