



A Unified Framework for Double Sweep Methods for the Helmholtz Equation

Nacime Bouziani, Frédéric Nataf, Pierre-Henri Tournier

► To cite this version:

Nacime Bouziani, Frédéric Nataf, Pierre-Henri Tournier. A Unified Framework for Double Sweep Methods for the Helmholtz Equation. 2022. hal-02964033v3

HAL Id: hal-02964033

<https://hal.science/hal-02964033v3>

Preprint submitted on 16 Oct 2022 (v3), last revised 4 Dec 2023 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Unified Framework for Double Sweep Methods for the Helmholtz Equation

Nacime Bouziani*, Frédéric Nataf and Pierre-Henri Tournier[†]

Contents

1	Introduction	1
2	Statement of the problem and two classical algorithms	2
3	Substructuring	3
3.1	Double sweep algorithm of [27, 30]	4
3.2	Three subdomain case	5
3.2.1	Subdomain wise (SW) ordering	5
3.2.2	Left-Right (LR) ordering	5
3.2.3	Dependence of the preconditioners on ordering	6
4	Analysis in the many subdomain case	6
4.1	Subdomain wise (SW) numbering	7
4.2	Left-Right (LR) numbering	7
4.3	Jacobi, Gauss-Seidel and Symmetric Gauss-Seidel for the subdomain wise numbering	8
4.4	Block Jacobi, Gauss-Seidel and Symmetric Gauss-Seidel for the left-right numbering	9
4.5	Theoretical comparison between the SGS and BJ preconditioners	11
4.6	Volumic preconditioners	11
5	Numerical results	13
5.1	Homogeneous waveguide	13
5.2	Influence of the overlap	13
5.3	Open cavity test	15
5.4	Marmousi	15
5.5	3D Overthrust	16
6	Conclusion	18

Abstract

We consider sweeping domain decomposition preconditioners to solve the Helmholtz equation in the case of stripwise domain decomposition with or without overlaps. We unify their derivation and convergence studies as Jacobi, Gauss-Seidel or Symmetric Gauss-Seidel for different numbering of the unknowns. This enables the theoretical comparisons of the double sweep methods in [23, 31] with that of [27, 30, 28]. It also makes possible the introduction of two new sweeping algorithms. We provide numerical test cases that assess the validity of the theoretical studies.

1 Introduction

Solving the Helmholtz equation numerically is a difficult task, especially when dealing with high-frequency regimes, heterogeneous media or reflecting boundary conditions. Over the last decades a lot of effort and progress has been made in developing efficient algorithms to solve the ill-conditioned linear system resulting from the Helmholtz operator's discretization. Domain decomposition methods (DDM) try to overcome these difficulties. They are hybrid methods that combine

*Department of Mathematics, Imperial College London, London, SW7 2AZ, UK (n.bouziani18@imperial.ac.uk)

[†] Laboratoire J.L. Lions, Université Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France, and ALPINES INRIA, Paris, France (frederic.nataf@sorbonne-universite.fr, pierre-henri.tournier@sorbonne-universite.fr)

direct solvers in subdomains and iterative matching of the solutions across the subdomains. The original domain decomposition method introduced by Schwarz [26] only works for overlapping domain decomposition. P. L. Lions [20] introduced a new variant of this algorithm where the Dirichlet interface conditions are replaced by Robin interface conditions, his method can be applied to both overlapping and nonoverlapping subdomains. He showed convergence for the elliptic case for a non overlapping domain decomposition. The proof was extended by Després [9] to the Helmholtz equation and later on to the time-harmonic Maxwell equations [10]. More recently, sweeping-type domain decomposition methods have been made popular due to their capability to achieve nearly-linear asymptotic complexity. A sweeping algorithm was first proposed and analyzed in [23] for convection-diffusion operators. Sweeping approaches for Helmholtz problems have recently seen their interest renewed as a preconditioner to speed up the convergence of the solver: the double sweep preconditioners of Stolk for overlapping decomposition [27, 28] and of Vion and Geuzaine for non overlapping decomposition with high order interface conditions [30, 31], the PML-based sweep method of Stolk [27], and the polarized traces method of Zepeda-Núñez and Demanet [32]. There also exists sweeping-type methods that are not domain decomposition based methods, such as the sweeping PML preconditioner of Engquist and Ying [13, 12], the source transfer method [5], see [15] for a complete panorama and relations between these methods.

The highlights of the article are:

- New formulation of [27, 30] which allows for two new variants introduced in § 4.4
- Unified convergence analysis for the three above mentioned algorithms which enables a comparison with the algorithms proposed in [10], [23, 31], see Table 1.
- Theoretical and numerical comparisons that show the advantage of the double sweep method in [23, 31] over the double sweep method in [27, 30].

We first state the problem in § 2. Then we explain in § 3 how to substructure the problem in terms of interface unknowns and how to apply classical linear algebra preconditioners (Jacobi, Gauss-Seidel and Symmetric Gauss-Seidel) to two different unknown numberings. After these preparatory tools have been introduced, we present and analyze in § 4 the convergence of six sweeping algorithms. Numerical results are shown in § 5.

2 Statement of the problem and two classical algorithms

We consider the Helmholtz equation in a bounded domain $\Omega \subset \mathbb{R}^2$ with frequency ω , velocity c and wavenumber k defined by $k = \omega/c$:

$$\begin{aligned} (-k^2 - \Delta) u &= f \text{ in } \Omega \\ &+ \text{appropriate boundary conditions on } \partial\Omega. \end{aligned} \quad (1)$$

We consider a layered decomposition of Ω into N slices $(\Omega_i)_{1 \leq i \leq N}$ with or without overlap, see Figure 1. The boundary $\partial\Omega_i \setminus \partial\Omega$ is written as the disjoint union of $\Gamma_{i,l}$ and $\Gamma_{i,r}$ where $\Gamma_{i,l}$ is on the left of Ω_i and $\Gamma_{i,r}$ is on its right ($\Omega_{1,l} = \emptyset$ and $\Omega_{N,r} = \emptyset$) (see Fig. 2). The outward normal from Ω_i on $\Gamma_{i,l}$ (resp. $\Gamma_{i,r}$) is denoted by $\vec{n}_{i,l}$ (resp. $\vec{n}_{i,r}$). The problem (1) can

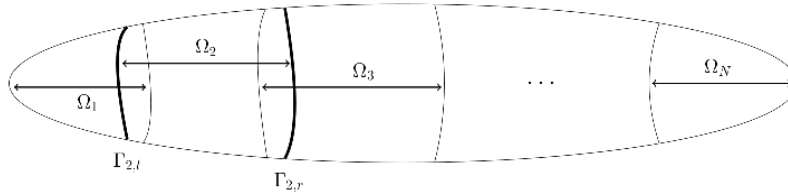


Figure 1: Decomposition into vertical strips

be solved iteratively using a domain decomposition method where we solve locally on each subdomain Ω_i the equation (1) with appropriate boundary conditions on the physical boundaries and interfaces [9]. The method writes:

Solve in parallel:

$$\begin{cases} (-k^2 - \Delta) u_i^{n+1} = f & \text{in } \Omega_i, \ 1 \leq i \leq N \\ \mathcal{B}_{i,l}(u_i^{n+1}) = \mathcal{B}_{i,l}(u_{i-1}^n) & \text{on } \Gamma_{i,l}, \ 2 \leq i \leq N \\ \mathcal{B}_{i,r}(u_i^{n+1}) = \mathcal{B}_{i,r}(u_{i+1}^n) & \text{on } \Gamma_{i,r}, \ 1 \leq i \leq N-1 \\ & + \text{appropriate boundary conditions on } \partial\Omega \cap \partial\Omega_i, \end{cases} \quad (2)$$

where $\mathcal{B}_{i,l}$ and $\mathcal{B}_{i,r}$ are the interface conditions. Here I denotes the square root of -1 ($I^2 = -1$). For sake of simplicity, we consider here either zeroth-order ABC (ABC0):

$$\begin{cases} \mathcal{B}_{i,l} = \partial_{\vec{n}_{i,l}} + Ik \\ \mathcal{B}_{i,r} = \partial_{\vec{n}_{i,r}} + Ik, \end{cases} \quad (3)$$

exact ABC as interface conditions or subdomains coupling via Perfectly Matched Layer (PML) as in [27]. In practice, when used as truncation conditions on artificial boundaries, ABC0 yields high non physical reflection of the order of 10% of the incoming wave whereas by definition exact ABC leads to no reflection at all. ABC0 boundary conditions are easy and cheap to use whereas exact ABC, always defined in theory, are sometimes impossible to use in practice (e.g. variable coefficients problems). As a result, there is room for compromise and a great deal of literature has been devoted to introduce various high order interface conditions. The most notable techniques are based on partial differential operators (see [11] and [2]) or on PML (see [19] and [6]), see [22] as well for a gentle introduction to this question. When used as truncation conditions, the final accuracy of the computation does depend on the choice of the ABC.

But let us stress that here, ABCs are used as interface conditions in domain decomposition methods so that the final accuracy of the computed result is not impacted by the choice of the ABC, only the iteration counts to solution are impacted. High-order ABC tailored to domain decomposition improve the iteration counts with respect to ABC0, see e.g. [14, 1]. A remarkable super convergence result noticed in [17] for the two subdomain case and in [24] for a decomposition into N strips (see Fig. 1) is that the use of exact ABCs as interface conditions yields convergence in a number of iterations equal to the number of subdomains. Since the solution in a subdomain depends on the value of the right hand side everywhere and that in algorithm (2) a subdomain receives data only from its neighbors it is not possible to achieve convergence in less than N iterations. When looking at the proof in [24], it appears that somehow the correct information flows from the extreme subdomain labelled 1 to the right and at the same time from subdomain labelled N to the left.

This motivated the search for algorithms which would sweep over the subdomains to reach convergence in one iteration consisting of a double sweep. In the sequel we will consider the double sweep algorithms introduced in [23, 31] and in [27, 30] since they converge in one double sweep if exact ABCs are used as interface conditions. Note that both algorithms were named double sweep algorithms which could be confusing. But as we shall see they are actually not the same and have different convergence rates when implemented with non exact ABCs. We start with the double sweep algorithm introduced in [23]. It consists in double sweeps over the subdomains:

left to right sweep:

$$\begin{cases} (-k^2 - \Delta) u_i^{n+1/2} = f \text{ in } \Omega_i, 1 \leq i \leq N \\ \mathcal{B}_{i,l} (u_i^{n+1/2}) = \mathcal{B}_{i,l} (u_{i-1}^{n+1/2}) \text{ on } \Gamma_{i,l}, 2 \leq i \leq N \\ \mathcal{B}_{i,r} (u_i^{n+1/2}) = \mathcal{B}_{i,r} (u_{i+1}^n) \text{ on } \Gamma_{i,r}, 1 \leq i \leq N-1 \\ + \text{ appropriate boundary conditions on } \partial\Omega \cap \partial\Omega_i, \end{cases} \quad (4)$$

then, right to left sweep:

$$\begin{cases} (-k^2 - \Delta) u_i^{n+1} = f \text{ in } \Omega_i, 1 \leq i \leq N \\ \mathcal{B}_{i,l} (u_i^{n+1}) = \mathcal{B}_{i,l} (u_{i-1}^{n+1/2}) \text{ on } \Gamma_{i,l}, 2 \leq i \leq N \\ \mathcal{B}_{i,r} (u_i^{n+1}) = \mathcal{B}_{i,r} (u_{i+1}^{n+1}) \text{ on } \Gamma_{i,r}, 1 \leq i \leq N-1 \\ + \text{ appropriate boundary conditions on } \partial\Omega \cap \partial\Omega_i. \end{cases} \quad (5)$$

It can be seen as a Symmetric Gauss-Seidel version of the Jacobi algorithm (2). This statement will be made more precise in section 4. Its study is made easier in its substructured formulation which is moreover needed to introduce the double sweep algorithm of [27, 30]. We devote the next section to substructuring.

3 Substructuring

In this section, we introduce the substructured problem related to algorithm (2). This will be the basis for the unified framework of DD sweeping methods. Substructuring consists in reformulating the iterative method considering only surface unknowns on the interfaces:

$$\begin{cases} h_{i,l}^n := \mathcal{B}_{i,l} (u_i^n), \text{ on } \Gamma_{i,l} \text{ for } 2 \leq i \leq N \\ h_{i,r}^n := \mathcal{B}_{i,r} (u_i^n), \text{ on } \Gamma_{i,r} \text{ for } 1 \leq i \leq N-1. \end{cases} \quad (6)$$

Considering the global vector h^n containing the local unknowns $(h_{i,l}^n)_{2 \leq i \leq N}$ and $(h_{i,r}^n)_{1 \leq i \leq N-1}$, we can reformulate the additive Schwarz method (2) as a Jacobi algorithm on h^n :

$$h^{n+1} := \mathcal{T}(h^n) + G, \quad (7)$$

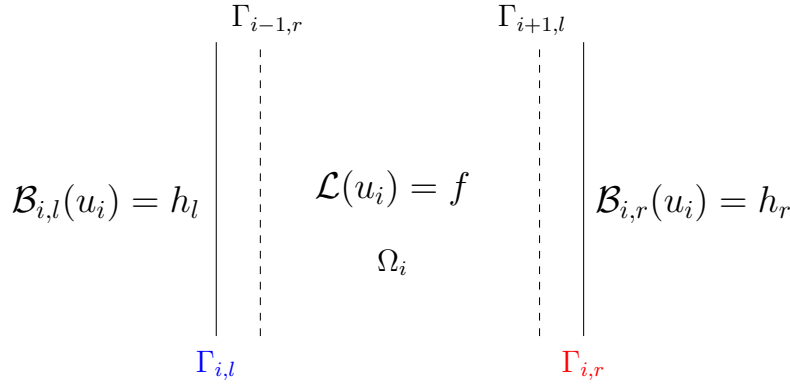


Figure 2: Local problem on the subdomain Ω_i

where the iteration operator $\mathcal{T}$ can be written in the form of an operator valued matrix and G refers to the contribution of the right-hand side f , see [24]. The above equation is what is called a substructured formulation of the volumic algorithm (2). Taking the limit as n tends to infinity, we see that we look for a vector h such that,

$$(Id - \mathcal{T})(h) = G. \quad (8)$$

Equation (8) is what is called the substructured formulation of the domain decomposition problem. In order to define more precisely the operator $\mathcal{T}$, we introduce for each subdomain an operator S_i which takes three arguments, two surface functions h_l and h_r and a volume function f and maps them to the local solution v :

$$S_i(h_l, h_r, f) := v \quad (9)$$

where $v : \Omega_i \mapsto \mathbb{C}$ satisfies:

$$\begin{cases} (-k^2 - \Delta) v = f & \text{in } \Omega_i \\ \mathcal{B}_{i,l}(v) = h_l & \text{on } \Gamma_{i,l} \quad (2 \leq i \leq N) \\ \mathcal{B}_{i,r}(v) = h_r & \text{on } \Gamma_{i,r} \quad (1 \leq i \leq N-1) \\ + \text{ appropriate boundary conditions on } \partial\Omega \cap \partial\Omega_i, \end{cases} \quad (10)$$

for $1 < i < N$. For $i = 1$, the definition of S_1 is similar except that it takes only the two arguments (h_r, f) since domain Ω_1 has no left interface and similarly operator S_N takes only the two arguments (h_l, f) since domain Ω_N has no right interface. **As of now, for sake of simplicity and by abuse of notation, $S_1(h_l, h_r, f)$ (resp. $S_N(h_l, h_r, f)$) will refer to $S_1(h_r, f)$ (resp. $S_N(h_l, f)$).**

Next, we introduce the surface right hand-side $G(f)$ by

$$\begin{aligned} G_{i,l}(f) &:= \mathcal{B}_{i,l}(S_{i-1}(0, 0, f)), \quad 2 \leq i \leq N \\ G_{i,r}(f) &:= \mathcal{B}_{i,r}(S_{i+1}(0, 0, f)), \quad 1 \leq i \leq N-1. \end{aligned} \quad (11)$$

and the substructured operator $\mathcal{T}$ by:

$$\begin{aligned} \mathcal{T}(h)_{i+1,l} &:= \mathcal{B}_{i+1,l}(S_i(h_{i,l}, h_{i,r}, 0)), \quad 1 \leq i \leq N-1 \\ \mathcal{T}(h)_{i-1,r} &:= \mathcal{B}_{i-1,r}(S_i(h_{i,l}, h_{i,r}, 0)), \quad 2 \leq i \leq N. \end{aligned} \quad (12)$$

The operator $\mathcal{T}$ has thus the following possibly non zero entries:

$$\begin{aligned} \mathcal{T}_{(i+1,l)(i,l)} &:= \mathcal{B}_{i+1,l}(S_i(\cdot, 0, 0)), \quad 1 \leq i \leq N-1 \\ \mathcal{T}_{(i+1,l)(i,r)} &:= \mathcal{B}_{i+1,l}(S_i(0, \cdot, 0)), \quad 1 \leq i \leq N-1 \\ \mathcal{T}_{(i-1,r)(i,r)} &:= \mathcal{B}_{i-1,r}(S_i(0, \cdot, 0)), \quad 2 \leq i \leq N \\ \mathcal{T}_{(i-1,r)(i,l)} &:= \mathcal{B}_{i-1,r}(S_i(\cdot, 0, 0)), \quad 2 \leq i \leq N. \end{aligned} \quad (13)$$

3.1 Double sweep algorithm of [27, 30]

The rationale behind the algorithm in [27, 30] is that for exact ABC used as interface conditions, we have a decoupling of left and right interface unknowns since

$$\mathcal{T}_{(i+1,l)(i,r)}^{EABC} \equiv 0 \text{ and } \mathcal{T}_{(i-1,r)(i,l)}^{EABC} \equiv 0. \quad (14)$$

Then, the operator $\mathcal{T}^{EABC}$ is nilpotent of order $N - 1$. This is related to the convergence in N iterations of algorithm (2) with exact ABC as interface conditions. In practice, ABC and even PML truncation techniques are not perfect and the nilpotency effect is lost. In order to force it, a new operator is introduced:

$$\begin{aligned}\mathcal{T}_{SDS}(h)_{i+1,l} &:= \mathcal{B}_{i+1,l}(S_i(h_{i,l}, 0, 0)), \quad 1 \leq i \leq N - 1 \\ \mathcal{T}_{SDS}(h)_{i-1,r} &:= \mathcal{B}_{i-1,r}(S_i(0, h_{i,r}, 0)), \quad 2 \leq i \leq N\end{aligned}\quad (15)$$

which is by construction nilpotent of order $N - 1$ even with non exact ABCs as interface conditions. A fixed point method based on this reads:

$$(Id - \mathcal{T}_{SDS})(h^{n+1}) = (\mathcal{T} - \mathcal{T}_{SDS})(h^n) + G. \quad (16)$$

More efficiently, the operator $I - \mathcal{T}_{SDS}$ can then be used as a preconditioner in order to solve the substructured problem (8). For instance the left preconditioned system

$$(Id - \mathcal{T}_{SDS})^{-1}(Id - \mathcal{T})(h) = (Id - \mathcal{T}_{SDS})^{-1}G \quad (17)$$

can be solved by a Krylov type method. A closer look at the operator $I - \mathcal{T}_{SDS}$ shows that inverting it can be made by two concurrent sweeps, hence the name double sweep.

In the sequel, algorithms other than (7) or (17) will be introduced as classical Jacobi, Gauss-Seidel, symmetric Gauss-Seidel applied to the substructured problem (8) with two different numberings of the interfaces. This will give a unified view to the methods considered in [8, 23, 27, 30, 31] and it will enable the introduction of two new algorithms in § 4.4.

3.2 Three subdomain case

For sake of clarity and to give a taste of the general case, we start with a three-domain decomposition of the whole plane $\mathbb{R}^2$, see Figure 3.

3.2.1 Subdomain wise (SW) ordering

The vector h has four components which are interfaces functions living respectively on $\Gamma_{1,r}$, $\Gamma_{2,l}$, $\Gamma_{2,r}$ and $\Gamma_{3,l}$, see Figure 3:

$$h := (h_{1,r} \ h_{2,l} \ h_{2,r} \ h_{3,l})^T.$$

This is the natural geometric ordering that we will refer to as the subdomain wise numbering (SW). Then for arbitrary interface conditions $\mathcal{B}_{i,l \text{ or } r}$, the sparsity pattern of the substructured problem is:

$$(I - \mathcal{T}_{SW})(h_{SW}) = \begin{pmatrix} I & X & X & 0 \\ X & I & 0 & 0 \\ 0 & 0 & I & X \\ 0 & X & X & I \end{pmatrix} \begin{pmatrix} h_{1,r} \\ h_{2,l} \\ h_{2,r} \\ h_{3,l} \end{pmatrix}, \quad (18)$$

where X denotes a possibly non zero entry.

3.2.2 Left-Right (LR) ordering

Following [23], we consider now an ordering where left interfaces are numbered first and then the right interfaces in the reverse order. In our three subdomain case, we define;

$$h := (h_{2,l} \ h_{3,l} \ h_{2,r} \ h_{1,r})^T.$$

For arbitrary interface conditions $\mathcal{B}_{i,l \text{ or } r}$, the sparsity pattern of the substructured operator is:

$$(I - \mathcal{T}_{LR})(h_{LR}) = \left(\begin{array}{cc|cc} I & 0 & 0 & X \\ X & I & X & 0 \\ \hline 0 & X & I & 0 \\ X & 0 & X & I \end{array} \right) \begin{pmatrix} h_{2,l} \\ h_{3,l} \\ \hline h_{2,r} \\ h_{1,r} \end{pmatrix}. \quad (19)$$

3.2.3 Dependence of the preconditioners on ordering

Of course, systems (18) and (19) are strictly equivalent. But as a consequence of the different numberings, the approximate inverses obtained from Jacobi or Gauss-Seidel type mechanisms may not be the same. As an example, consider the Gauss-Seidel preconditioner for the SW numbering:

$$(I - \mathcal{T}_{SW,GS})(h_{SW}) := \begin{pmatrix} I & 0 & 0 & 0 \\ \mathcal{T}_{(2,l)(1,r)} & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & \mathcal{T}_{(3,l)(2,l)} & \mathcal{T}_{(3,l)(2,r)} & I \end{pmatrix} \begin{pmatrix} h_{1,r} \\ h_{2,l} \\ h_{2,r} \\ h_{3,l} \end{pmatrix}, \quad (20)$$

and the Gauss-Seidel preconditioner for the LR numbering:

$$(I - \mathcal{T}_{LR,GS})(h_{LR}) := \begin{pmatrix} I & 0 & 0 & 0 \\ \mathcal{T}_{(3,l)(2,l)} & I & 0 & 0 \\ 0 & \mathcal{T}_{(2,r)(3,l)} & I & 0 \\ \mathcal{T}_{(1,r)(2,l)} & 0 & \mathcal{T}_{(1,r)(2,r)} & I \end{pmatrix} \begin{pmatrix} h_{2,l} \\ h_{3,l} \\ h_{2,r} \\ h_{1,r} \end{pmatrix}. \quad (21)$$

Even up to a reordering these two preconditioners are different since the entries of $\mathcal{T}$ which are kept in these two approximations are not the same. In § 4, we will develop this analysis for other preconditioners with also an arbitrary number of subdomains.

Another important fact already mentioned above, see eq. (14), is that if the interface conditions are exact absorbing boundary conditions (EABC), four additional entries cancel in (18) or (19) so that $\mathcal{T}_{SW}$ and $\mathcal{T}_{LR}$ are actually nilpotent operators of order 2. Indeed the entry $\mathcal{T}_{(1,r)(2,l)}$ is the operator that maps $h_{2,l}$ to $\mathcal{B}_{1,r}(v_{2,l})$ where $v_{2,l}$ satisfies:

$$\begin{aligned} \mathcal{L}(v_{2,l}) &= 0 \text{ in } \Omega_2, \\ \mathcal{B}_{2,r}(v_{2,l}) &= 0 \text{ on } \Gamma_{2,r}, \\ \mathcal{B}_{2,l}(v_{2,l}) &= h_{2,l} \text{ on } \Gamma_{2,l}. \end{aligned} \quad (22)$$

Since $\mathcal{B}_{2,r}$ is an EABC, $v_{2,l}$ can be seen as the restriction of a harmonic function defined on the half plane at the right of $\Gamma_{2,l}$ so that $\mathcal{B}_{1,r}(v_{2,l}) = 0$ since $\mathcal{B}_{1,r}$ is an EABC as well. That is, $\mathcal{T}_{(1,r)(2,l)} = 0$. In the same way, we can prove three other cancellations, namely: $\mathcal{T}_{(2,r)(3,l)} = 0$ and $\mathcal{T}_{(i+1,l)(i,r)} = 0$ for $i = 1, 2$. The only entries left are $\mathcal{T}_{(i,r)(i+1,r)}$ and $\mathcal{T}_{(i+2,l)(i+1,l)}$ for $i = 1$. Let us denote by $\mathcal{T}_{SW}^{EABC}$ the operator $\mathcal{T}_{SW}$ when the interface conditions are EABC. We have just proved that its sparsity pattern is then:

$$\mathcal{T}_{SW}^{EABC} = \begin{pmatrix} 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 \end{pmatrix}. \quad (23)$$

As for the left-right numbering, let us denote by $\mathcal{T}_{LR}^{EABC}$ the operator $\mathcal{T}_{LR}$ when the interface conditions are EABC so that with four entries cancelled, its sparsity pattern is then:

$$\mathcal{T}_{LR}^{EABC} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ X & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & X & 0 \end{pmatrix}. \quad (24)$$

Another way to see that numbering impacts the preconditioners is to remark that even for EABCs, the Gauss-Seidel preconditioner for the left-right numbering is exact:

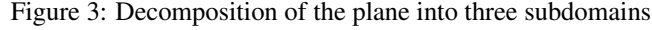
$$(I - \mathcal{T}_{LR,GS}^{EABC}) = (I - \mathcal{T}_{LR}^{EABC}),$$

which is not the case for the Gauss-Seidel preconditioner for the subdomain wise numbering

$$(I - \mathcal{T}_{SW,GS}^{EABC}) \neq (I - \mathcal{T}_{SW}^{EABC}).$$

4 Analysis in the many subdomain case

We come back to the case of a stripwise decomposition into N subdomains as in Figures 1 or 2. As in the three subdomain case, we consider two numberings for the substructured system (8).



$$\boxed{\begin{aligned} \mathcal{M}_r^{N-1} = \mathcal{M}_l^{N-1} = 0; \quad \mathcal{M}_l \mathcal{M}_r = \mathcal{M}_r \mathcal{M}_l = 0; \quad \mathcal{A}_l^2 = \mathcal{A}_r^2 = 0 \\ \mathcal{A}_l \mathcal{M}_l = \mathcal{A}_r \mathcal{M}_r = 0; \quad \mathcal{M}_l \mathcal{A}_r = \mathcal{M}_r \mathcal{A}_l = 0. \end{aligned}} \quad (27)$$

It is worth noticing that these relations come from the structure of the matrices and do not depend on the value of the entries. Let us introduce the following operators that will play a crucial role in the sequel:

$$C_r := (I - \mathcal{M}_r)^{-1} \mathcal{A}_r = \left(\sum_{i=0}^{N-2} \mathcal{M}_r^i \right) \mathcal{A}_r \text{ and } C_l := (I - \mathcal{M}_l)^{-1} \mathcal{A}_l = \left(\sum_{i=0}^{N-2} \mathcal{M}_l^i \right) \mathcal{A}_l. \quad (28)$$

Note that using cancellation relations (27), we have

$$C_r^2 = C_l^2 = 0. \quad (29)$$

4.3 Jacobi, Gauss-Seidel and Symmetric Gauss-Seidel for the subdomain wise numbering

With the notations introduced above, the substructured operator (25) reads:

$$(I - \mathcal{T}_{SW}) = \begin{bmatrix} I & -\mathcal{A}_{r,2N-2,1} & -\mathcal{M}_{r,2N-2,2N-3} & 0 & & & \\ -\mathcal{A}_{l,1,2N-2} & I & 0 & 0 & & & \\ 0 & 0 & I & -\mathcal{A}_{r,2N-3,2} & -\mathcal{M}_{r,2N-3,2N-4} & 0 & \\ & \mathcal{M}_{l,2,1} & \mathcal{A}_{l,2,2N-3} & I & 0 & 0 & \\ & & 0 & 0 & \ddots & \ddots & \ddots \\ & & & -\mathcal{M}_{l,3,2} & \ddots & \ddots & \ddots \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \ddots & \ddots & I \end{bmatrix}. \quad (30)$$

When considering the decomposition of matrix $I - \mathcal{T}_{SW}$ into its diagonal, lower and upper parts as $I - \mathcal{T}_{SW} = L + D + U$, its upper part U involves only entries of the matrices $\mathcal{A}_r$ and $\mathcal{M}_r$ and similarly, its lower part L involves only entries of the matrices $\mathcal{A}_l$ and $\mathcal{M}_l$ and the diagonal part D is simply the identity. Expressed now in the LR numbering, it is then easy to check that we have $D = I$, $L = -\mathcal{M}_l - \mathcal{A}_l$ and $U = -\mathcal{M}_r - \mathcal{A}_r$. Thus we introduce in a classical way several preconditioners whose formulas are given in the LR numbering:

- the Jacobi preconditioner D^{-1} :

$$M_{Jacobi}^{-1} := I,$$

- the left Gauss-Seidel (GS) preconditioner $(L + D)^{-1}$:

$$M_{GS}^{-1} := (I - \mathcal{M}_l - \mathcal{A}_l)^{-1},$$

- the Symmetric Gauss-Seidel (SGS) preconditioner $[(L + D)D^{-1}(D + U)]^{-1} = (D + U)^{-1}D(L + D)^{-1}$:

$$M_{SGS}^{-1} := (I - \mathcal{M}_r - \mathcal{A}_r)^{-1}(I - \mathcal{M}_l - \mathcal{A}_l)^{-1}.$$

The Jacobi preconditioner amounts to Desprès-Lions algorithm and the two other preconditioners had been introduced and studied in [23, 31]. In [23], the GS preconditioner was referred to as FDA and the SGS preconditioner as the Double Sweep (DS) algorithm.

When used as solvers, their convergence depends on the spectral radius of the error propagation operators defined as:

Definition 4.1. For a preconditioner M^{-1} of a linear operator A , we denote by R its error propagation operator as: $R := M^{-1}(M - A)$.

It was proved in [23] that the spectral radius of their error propagation operators denoted by $\rho(R)$ satisfy the following bounds: $\rho(R_{Jacobi}) \leq \rho^{1/N}$, $\rho(R_{GS}) \leq \rho^{2/N}$ and $\rho(R_{SGS}) \leq \rho$ where N is the number of subdomains and $\rho = \rho(C_r)\rho(C_l)$. The proof was written with the help of the formal language theory, see e.g. [21]. The SGS algorithm can be studied more classically since we have:

Proposition 4.2. The formula for R_{SGS} is:

$$R_{SGS} = (I + C_r)C_l(\mathcal{M}_r + \mathcal{A}_r).$$

As for the power of R_{SGS} , we have the following formula:

$$R_{SGS}^n = (I + C_r)C_l(C_r C_l)^{n-1}(\mathcal{M}_r + \mathcal{A}_r).$$

Proof. By definition, we have:

$$R_{SGS} = (Id - \mathcal{A}_r - \mathcal{M}_r)^{-1} (Id - \mathcal{M}_l - \mathcal{A}_l)^{-1} (\mathcal{M}_l + \mathcal{A}_l) (\mathcal{A}_r + \mathcal{M}_r).$$

Using the equalities: $\mathcal{M}_l \mathcal{A}_r = \mathcal{M}_l \mathcal{M}_r = 0$ and

$$Id - \mathcal{A}_r - \mathcal{M}_r = (Id - \mathcal{A}_r) (Id - \mathcal{M}_r),$$

we have:

$$R_{SGS} = ((Id - \mathcal{M}_r)^{-1} + C_r) C_l (\mathcal{A}_r + \mathcal{M}_r).$$

Note that since $\mathcal{M}_r \mathcal{M}_l = \mathcal{M}_r \mathcal{A}_l = 0$, we have $\mathcal{M}_r C_l = 0$ and thus $(Id - \mathcal{M}_r)^{-1} C_l = C_l$.

As for the powers of R_{SGS} , we first note that:

$$R_{SGS}^n = (I + C_r) C_l [(\mathcal{M}_r + \mathcal{A}_r)(I + C_r) C_l]^{n-1} (\mathcal{M}_r + \mathcal{A}_r).$$

Using cancellation relations (27) to simplify the middle term, we get:

$$\begin{aligned} (\mathcal{M}_r + \mathcal{A}_r)(I + C_r) C_l &= \left[\left(\sum_{i=1}^{N-2} \mathcal{M}_r^i \right) \mathcal{A}_r + \mathcal{M}_r + \mathcal{A}_r \right] C_l \\ &= C_r C_l, \end{aligned}$$

where the last equality comes from $\mathcal{M}_r C_l = 0$. □

4.4 Block Jacobi, Gauss-Seidel and Symmetric Gauss-Seidel for the left-right numbering

Considering system (26) as a block two by two matrix, the matrix $I - \mathcal{T}_{LR}$ can be decomposed into its block diagonal, lower and upper parts as $L + D + U$ where $D = I - \mathcal{M}_r - \mathcal{M}_l$, $L = -\mathcal{A}_r$ and $U = -\mathcal{A}_l$. Thus we introduce in a classical way several preconditioners:

- the Block Jacobi preconditioner (BJ) D^{-1}
- the Block Gauss-Seidel (BGS) preconditioner $(L + D)^{-1}$
- the Block Symmetric Gauss-Seidel (BSGS) preconditioner $[(L + D)D^{-1}(D + U)]^{-1} = (D + U)^{-1} D (L + D)^{-1}$

Remark : The BJ preconditioner was introduced in [27, 28] for overlapping decompositions and in [30] for non overlapping decompositions. Note that a diagonal block solve consists in two independent forward substitutions on the left and right interface unknowns. So the method was called Double Sweep although it is different from the one previously introduced in [23] and named as well Double Sweep. We hope to have clarified this possible confusion.

We have thus three preconditioners with the following formulas:

$$\begin{aligned} M_{BJ}^{-1} &:= (I - \mathcal{M}_r - \mathcal{M}_l)^{-1}, \\ M_{BGS}^{-1} &:= (I - \mathcal{M}_r - \mathcal{M}_l - \mathcal{A}_r)^{-1}, \\ M_{BSGS}^{-1} &:= (I - \mathcal{M}_r - \mathcal{M}_l - \mathcal{A}_l)^{-1} (I - \mathcal{M}_r - \mathcal{M}_l) (I - \mathcal{M}_r - \mathcal{M}_l - \mathcal{A}_r)^{-1}. \end{aligned} \tag{31}$$

To our knowledge, the last two algorithms have not been introduced before (except for the BGS algorithm mentioned briefly in [27] eq. (17-23), page 246). Their convergence rates nor that of BJ had not been studied before. They arise from our way to introduce the BJ algorithm which is different from the one developed in [27, 30] and that has been recalled above in § 3.1.

We have the following propositions.

Proposition 4.3. *The error propagation operator R_{BJ} has the following expression:*

$$R_{BJ} = C_r + C_l,$$

and for n even we have:

$$R_{BJ}^n = (C_r C_l)^{n/2} + (C_l C_r)^{n/2}.$$

Proof. Thanks to relations (27), we have $M_{BJ}^{-1} = Id + \sum_{i=1}^{N-2} \mathcal{M}_l^i + \sum_{i=1}^{N-2} \mathcal{M}_r^i$ and:

$$R_{BJ} = (Id + \sum_{i=1}^{N-2} \mathcal{M}_l^i + \sum_{i=1}^{N-2} \mathcal{M}_r^i) (\mathcal{A}_l + \mathcal{A}_r) = \sum_{i=0}^{N-2} \mathcal{M}_r^i \mathcal{A}_r + \sum_{i=0}^{N-2} \mathcal{M}_l^i \mathcal{A}_l = C_r + C_l.$$

Next using cancellation relation (29), the formula for R_{BJ}^n can easily be proved by induction. □

Proposition 4.4. *The error propagation operator R_{BGS} has the following expression:*

$$R_{BGS} = (I + C_r) C_l ,$$

and for any integer n we have:

$$R_{BGS}^n = (I + C_r) C_l (C_r C_l)^{n-1} .$$

Proof. Using the following formulas:

$$\begin{aligned} (I - \mathcal{M}_r - \mathcal{M}_l)^{-1} &= (I - \mathcal{M}_r)^{-1} (I - \mathcal{M}_l)^{-1} = (I - \mathcal{M}_l)^{-1} (I - \mathcal{M}_r)^{-1} \\ (I - \mathcal{M}_r - \mathcal{M}_l - \mathcal{A}_r)^{-1} &= (I - \mathcal{M}_r)^{-1} (I + \mathcal{A}_r) (I - \mathcal{M}_l)^{-1} \\ (I - \mathcal{M}_r - \mathcal{M}_l - \mathcal{A}_l)^{-1} &= (I - \mathcal{M}_l)^{-1} (I + \mathcal{A}_l) (I - \mathcal{M}_r)^{-1} , \end{aligned} \quad (32)$$

in addition to (27), we have:

$$\begin{aligned} R_{BGS} &= M_{BGS}^{-1} \mathcal{A}_l = (I - \mathcal{M}_r)^{-1} (I + \mathcal{A}_r) (I - \mathcal{M}_l)^{-1} \mathcal{A}_l \\ &= (I - \mathcal{M}_r)^{-1} (I - \mathcal{M}_l)^{-1} \mathcal{A}_l + (I - \mathcal{M}_r)^{-1} \mathcal{A}_r (I - \mathcal{M}_l)^{-1} \mathcal{A}_l \\ &= (I - \mathcal{M}_l)^{-1} (I - \mathcal{M}_r)^{-1} \mathcal{A}_l + C_r C_l = C_l + C_r C_l . \end{aligned}$$

As for the n -th power of R_{BGS} , the formula can be proved by induction using the fact that:

$$(I + C_r) C_l (I + C_r) C_l = (I + C_r) C_l C_r C_l .$$

□

Proposition 4.5. *The error propagation operator R_{BSGS} has the following expression:*

$$R_{BSGS} = (I + C_l) C_r C_l ,$$

and for any integer n we have:

$$R_{BSGS}^n = (I + C_l) (C_r C_l)^n .$$

Proof. Using cancellation relations (27) and (32), we have:

$$\begin{aligned} R_{BSGS} &= M_{BGS}^{-1} L D^{-1} U = M_{BGS}^{-1} \mathcal{A}_r (I - \mathcal{M}_r)^{-1} (I - \mathcal{M}_l)^{-1} \mathcal{A}_l = M_{BGS}^{-1} \mathcal{A}_r C_l \\ &= (I - \mathcal{M}_l)^{-1} (I + \mathcal{A}_l) (I - \mathcal{M}_r)^{-1} (I - \mathcal{M}_l) (I - \mathcal{M}_r)^{-1} (I + \mathcal{A}_r) (I - \mathcal{M}_l)^{-1} \mathcal{A}_r C_l \\ &= (I - \mathcal{M}_l)^{-1} (I + \mathcal{A}_l - \mathcal{M}_l) (I - \mathcal{M}_r)^{-1} (I + \mathcal{A}_r) (I - \mathcal{M}_l)^{-1} \mathcal{A}_r C_l \\ &= (I + C_l) (I - \mathcal{M}_r)^{-1} (I + \mathcal{A}_r) \mathcal{A}_r C_l = (I + C_l) C_r C_l . \end{aligned}$$

As for the last formula of the proposition, it comes from the nullity of both C_r^2 and C_l^2 . □

Remark 4.1. *It is clear from the above results that the operators C_r and C_l (see eq. (28)) are a key measure of the efficiency of these algorithms. If the left interface conditions are exact absorbing conditions, operator $\mathcal{A}_r = 0$, see equation (22) and thus $C_r = 0$ as well. Similarly if right interface conditions are exact absorbing conditions, operator $C_l = 0$. More generally, the norms of C_r and C_l are proportional to that of $\mathcal{A}_r = 0$ and $\mathcal{A}_l = 0$. Thus as expected the more absorbing the interface conditions are, the better the convergence is. Another parameter is the number of subdomains since for given interface conditions, as the number of subdomains increases, the norm of C_r and C_l will grow. It echoes what was noticed in [27] at the end of § 4:*

”With the Robin transmission conditions the iteration numbers grow roughly linearly in N_x , or as $N^{1/2}$ in 2-D.”

In this respect, note that R_{BSGS} satisfies

$$(I + C_l)^{-1} R_{BSGS} (I + C_l) = C_r C_l$$

so that from all algorithms listed in Table 1 it has the most favorable amplification error operator iff the norm of $C_r C_l$ is smaller than one. In a recent article [16], these norms have been estimated in the context of the study of the convergence rate of the Jacobi preconditioner.

4.5 Theoretical comparison between the SGS and BJ preconditioners

We devote a paragraph to this comparison since these algorithms have been misleadingly coined the same name Double Sweep algorithm in the articles in which they had been introduced, in [23, 31] for SGS and in [27, 30] for BJ. From Propositions 4.2 and 4.3, we have for n even:

$$R_{BJ}^n = (C_r C_l)^{n/2} + (C_l C_r)^{n/2},$$

and for any n

$$R_{SGS}^n = (I + C_r) C_l (C_r C_l)^{n-1} (\mathcal{M}_r + \mathcal{A}_r).$$

Let us denote by

$$\rho \text{ the spectral radius of } C_r C_l, \quad (33)$$

which is the same as that of $C_l C_r$. We have for any operator norm $\|\cdot\|$ on the matrices the following estimates:

$$\|R_{BJ}^n\| \leq \|(C_r C_l)^{n/2}\| + \|(C_l C_r)^{n/2}\| \leq \|(C_r C_l)^{n/2}\| + \|C_l\| \|(C_r C_l)^{n/2-1}\| \|C_l\|, \quad (34)$$

so that taking the n -th square root, we get the following estimate for the spectral radius of R_{BJ} :

$$\rho(R_{BJ}) \leq \rho^{1/2}. \quad (35)$$

As for the SGS algorithm, we get:

$$\|R_{SGS}^n\| \leq \|I + C_r\| \|C_l\| \|\mathcal{M}_r + \mathcal{A}_r\| \|(C_r C_l)^{n-1}\|. \quad (36)$$

We thus get the following estimate for the spectral radius of R_{SGS} :

$$\rho(R_{SGS}) \leq \rho. \quad (37)$$

Thus when $\rho < 1$, an advantage of SGS over BJ is the square root of ρ in the ratio of the convergence rates. This is coherent with the factor two in the iteration counts in favour of SGS observed in most tables of the numerical section 5 when PMLs are interface conditions. Nevertheless, note that at the expense of doubling the number of cores used in the application of the BJ preconditioner, this difference is nullified in terms of elapsed time. Indeed, during one iteration of the SGS algorithm, only one subdomain is active at a time in the order $1 \rightarrow 2 \rightarrow \dots \rightarrow N$ and then in the reverse order $N \rightarrow N-1 \rightarrow \dots \rightarrow 1$. Whereas during one iteration of BJ, two subdomains are active at a time in the order $1 \rightarrow 2 \rightarrow \dots \rightarrow N$ for one subdomain and concurrently in the reverse order $N \rightarrow N-1 \rightarrow \dots \rightarrow 1$ for the other one. Thus, the elapsed time of one iteration of SGS is twice as much as one iteration of BJ using two cores. This compensates for the higher iteration counts of BJ compared to SGS when the number of cores is not limited.

Ref.	Abbr.	Linear Algebra	Definition	Ampl. Error	Radius
Després (1991)	Jacobi	Jacobi-SW	I	$\mathcal{M}_r + \mathcal{M}_l + \mathcal{A}_r + \mathcal{A}_l$	$\rho^{1/N}$
Nataf-Nier (1997)	GS	Gauss-Seidel-SW	$(I - \mathcal{M}_l - \mathcal{A}_l)^{-1}$	$(I + C_l)(\mathcal{M}_r + \mathcal{A}_r)$	$\rho^{2/N}$
{ Nataf-Nier (1997) Vion-Geuzaine(2018)	SGS	Symm. Gauss-Seidel-SW	$(I - \mathcal{M}_r - \mathcal{A}_r)^{-1} \times (I - \mathcal{M}_l - \mathcal{A}_l)^{-1}$	$(I + C_r) C_l (\mathcal{M}_r + \mathcal{A}_r)$	ρ
{ Stolk (2013) Vion-Geuzaine (2014)	BJ	Block Jacobi-LR	$(I - \mathcal{M}_r - \mathcal{M}_l)^{-1}$	$C_r + C_l$	$\rho^{1/2}$
Here (2022)	BGS	Block GS-LR	$(I - \mathcal{M}_r - \mathcal{M}_l - \mathcal{A}_r)^{-1}$	$(I + C_r) C_l$	ρ
Here (2022)	BSGS	Symm. Block GS-LR	$(I - \mathcal{M}_r - \mathcal{M}_l - \mathcal{A}_l)^{-1} \times (I - \mathcal{M}_r - \mathcal{M}_l) \times (I - \mathcal{M}_r - \mathcal{M}_l - \mathcal{A}_r)^{-1}$	$(I + C_l) C_r C_l$	ρ

Table 1: Algorithms and their convergence properties. SW means subdomain wise numbering and LR means left right numbering

4.6 Volumic preconditioners

The substructured algorithms demand exact local solves and data structures adapted to substructured algorithms. In order to allow for the use of fast approximate solvers in subdomains and of volumic data structures, an extension to the volumic case

is defined below. For the SGS algorithm, the volumic formulation is simply given by equations (4)-(5) followed by the use of a partition of unity in order to iterate on functions which are uniquely defined in the overlaps. For this, we first introduce for $1 \leq i \leq N$ partition of unity functions $\chi_i : \Omega_i \rightarrow \mathbb{R}^+$ and for $v_i : \Omega_i \rightarrow \mathbb{C}$, $E_i(v_i)$ denotes the extension by zero of v_i to Ω so that for any function $v : \Omega \rightarrow \mathbb{C}$, we have:

$$v = \sum_{i=1}^N E_i(\chi_i v|_{\Omega_i}).$$

We introduce the operator MC (MakeCoherent) that maps a collection of local functions $(v_i)_{1 \leq i \leq N}$ to a global function defined as follows:

$$MC((v_i)_{1 \leq i \leq N}) := \sum_{i=1}^N E_i(\chi_i v_i). \quad (38)$$

Let f be a source term and h be interface sources, it is then natural to introduce the following linear parallel reconstruction algorithm Vol:

$$\text{Vol}(h, f) := MC((S_i(h_{i,l}, h_{i,r}, f))_{1 \leq i \leq N}). \quad (39)$$

Let u be the solution to the original problem (1), we have:

$$u = \text{Vol}((I - \mathcal{T})^{-1}G(f), f). \quad (40)$$

Since a substructured preconditioner M^{-1} may be seen as an approximate inverse to $(I - \mathcal{T})$, it is natural to define the related volumic preconditioner as:

Definition 4.6. Let M^{-1} be a substructured preconditioner for problem (8), the related volumic preconditioner to problem (1) M_{vol}^{-1} is defined as follows:

$$M_{vol}^{-1}(f) := \text{Vol}(M^{-1}G(f), f) = \text{Vol}(M^{-1}G(f), 0) + \text{Vol}(0, f). \quad (41)$$

Note that if the surface variant of the preconditioner is exact as it happens for some if the interface conditions are exact absorbing boundary conditions, this property will be inherited by its volumic counterpart.

As an example, we detail below after some rewriting the action of the volumic preconditioner associated to the surface variant of the BJ preconditioner.

Volumic BJ Preconditioner

Left to right sweep

Subdomain 1

$$w_1 \leftarrow S_1(0, 0, f)$$

Subdomain 2

$$G_{2,l}(f) \leftarrow \mathcal{B}_{2,l}(w_1)$$

$$h_{2,l} \leftarrow G_{2,l}(f)$$

$$v_{2,l} \leftarrow S_2(h_{2,l}, 0, 0)$$

$$w_2 \leftarrow S_2(0, 0, f)$$

Subdomain 3

$$G_{3,l}(f) \leftarrow \mathcal{B}_{3,l}(w_2)$$

$$h_{3,l} \leftarrow G_{3,l}(f) + \mathcal{B}_{3,l}(v_{2,l})$$

$$v_{3,l} \leftarrow S_3(h_{3,l}, 0, 0)$$

$$w_3 \leftarrow S_3(0, 0, f)$$

Subdomain 4

$$G_{4,l}(f) \leftarrow \mathcal{B}_{4,l}(w_3)$$

$$h_{4,l} \leftarrow G_{4,l}(f) + \mathcal{B}_{4,l}(v_{3,l})$$

$\vdots$

Subdomain N

$$w_N \leftarrow S_N(0, 0, f)$$

$$v_{N,l} \leftarrow S_N(h_{N,l}, 0, 0)$$

Right to left sweep

Subdomain N

$$w_N \leftarrow S_N(0, 0, f)$$

Subdomain $N - 1$

$$G_{N-1,r}(f) \leftarrow \mathcal{B}_{N-1,r}(w_N)$$

$$h_{N-1,r} \leftarrow G_{N-1,r}(f)$$

$$v_{N-1,r} \leftarrow S_{N-1}(0, h_{N-1,r}, 0)$$

$$w_{N-1} \leftarrow S_{N-1}(0, 0, f)$$

Subdomain $N - 2$

$$G_{N-2,r}(f) \leftarrow \mathcal{B}_{N-2,r}(w_{N-1})$$

$$h_{N-2,r} \leftarrow G_{N-2,r}(f) + \mathcal{B}_{N-2,r}(v_{N-1,r})$$

$$v_{N-2,r} \leftarrow S_{N-2}(0, h_{N-2,r}, 0)$$

$$w_{N-2} \leftarrow S_{N-2}(0, 0, f)$$

Subdomain $N - 3$

$$G_{N-3,r}(f) \leftarrow \mathcal{B}_{N-3,r}(w_{N-2})$$

$$h_{N-3,r} \leftarrow G_{N-3,r}(f) + \mathcal{B}_{N-3,r}(v_{N-2,r})$$

$\vdots$

Subdomain 1

$$w_1 \leftarrow S_1(0, 0, f)$$

$$v_{1,r} \leftarrow S_1(h_{1,r}, 0, 0)$$

Then we perform local summations:

$$u_i := w_i + v_{i,l} + v_{i,r}, \quad 2 \leq i \leq N - 1$$

and

$$u_1 := w_1 + v_{1,r} \quad \text{and} \quad u_N := w_N + v_{N,l}.$$

Finally the action of the preconditioner $M_{BJ,vol}^{-1}$ reads:

$$M_{BJ,vol}^{-1}(f) := \sum_{i=1}^N E_i(\chi_i(u_i)). \quad (42)$$

5 Numerical results

In this section, we present numerical results for the volumic variants of the preconditioners of Table 1 used with the GMRES algorithm [25] with two different relative tolerances, $TOL=10^{-6}$ and $TOL=10^{-3}$. The Helmholtz equation is discretized with P1 or P2 finite elements using the FreeFem++ domain specific language [18]. The following test cases are shown: homogeneous waveguide (§ 5.1 and 5.2), open cavity (§ 5.3), Marmousi (§ 5.4) and 3D Overthrust (§ 5.5).

We also compare the use of zeroth-order ABC with that of Perfectly Matched Layers (PML) as interface conditions for the various sweeping algorithms. It is worth noting that in order to avoid transmitting attenuated solutions between subdomains, the width of the overlap region needs to be taken larger than the width of the PML. In the following numerical results, the overlap δ is equal to 4 mesh elements and the width of the PML is 2. The PML is defined following [3]. As we can see, using PML interface conditions, iteration counts are slightly better for all cases and much better for the Marmousi test case. Also note that as the number of subdomains increases, the superiority of PML becomes more pronounced.

When comparing the various algorithms, we see that the iteration counts are qualitatively in agreement with the spectral radius estimates summarized in Table 1 but only when using PML as interface conditions. For the Jacobi method, the increase is linear w.r.t. to the number of subdomains. The sweeping methods have iteration counts that increase sublinearly with the number of subdomains. When using PML, the Block Jacobi (BJ) methods needs twice as many iterations as the other sweeping methods. We also see that the two new preconditioners BGS and BSGS offer no iteration count improvement over the SGS preconditioner although they are more expensive.

5.1 Homogeneous waveguide

First, we consider the homogeneous waveguide test case with a layered decomposition into N subdomains. More specifically, we consider a rectangular geometry ($\Omega = [0, N] \times [0, 1]$) made of a homogeneous medium. On the upper and lower sides of the waveguide, we impose homogeneous Dirichlet conditions (cf. black lines in Figure 4). In addition, we perform a multimode excitation on the left side and we impose an absorbing boundary condition on the right side. The global problem is written as

$$\begin{cases} (-k^2 - \Delta) u = f & \text{in } \Omega \\ (\partial_{\bar{n}} + Ik) u = 0 & \text{on } \{x = N\} \times [0, 1] \\ (\partial_{\bar{n}} + Ik) u = u_g & \text{on } \{x = 0\} \times [0, 1] \\ u = 0 & \text{on } [0, N] \times \{y = 0, y = 1\} \end{cases} \quad (43)$$

where $u_g = e^{-120(y-0.5)^2} \sin(\pi y)$.

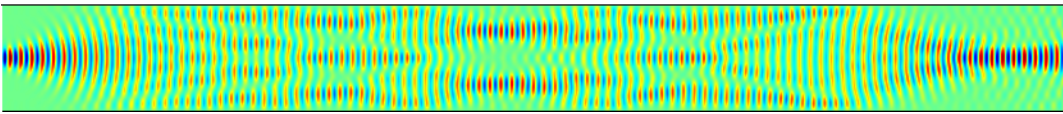


Figure 4: Homogeneous waveguide ($k = 20\pi$)

We considered two values for the wave number: $k = 20$ and $k = 20\pi$ and results are given in Tables 2 and 4 for the zeroth order interface conditions and in Tables 3 and 5 for PML interface conditions. The iteration counts of the various sweeping algorithms are in agreement with the spectral radius estimates of Table 1. In this simple case, the superiority of PML over zeroth order ABC is not so significant except for large number of subdomains.

5.2 Influence of the overlap

We have also tested the effect of the width of the overlap on the convergence in the case of the homogeneous waveguide with zeroth-order ABC. In this case, there is little if no effect of the overlap as shown in Table 6 where the overlap varies from 2 mesh sizes up to 16 mesh sizes.

N	Jacobi	BJ	BGS	BSGS	SGS
5	29 (18)	9 (5)	5 (3)	5 (3)	5 (3)
10	62 (39)	12 (7)	7 (4)	6 (4)	6 (4)
20	135 (81)	18 (10)	10 (6)	9 (6)	9 (6)
40	283 (163)	26 (12)	14 (8)	14 (7)	14 (8)
80	744 (329)	42 (20)	23 (12)	22 (12)	23 (12)

Table 2: Volumic preconditioner, homogeneous waveguide with zeroth-order ABC, $k = 20$, $\delta = 4h$, $\text{TOL}=10^{-6}(10^{-3})$, $\text{nppwl} = 24$, P1

N	Jacobi	BJ	BGS	BSGS	SGS
5	33 (21)	9 (5)	5 (2)	4 (2)	5 (3)
10	71 (40)	11 (6)	6 (3)	5 (3)	6 (3)
20	150 (80)	13 (7)	7 (4)	6 (3)	7 (4)
40	293 (143)	17 (9)	9 (5)	8 (5)	9 (5)
80	690 (276)	22 (14)	12 (7)	12 (7)	12 (7)

Table 3: Volumic preconditioner with PML interface conditions, homogeneous waveguide, $k = 20$, $\delta = 4h$, $\text{TOL}=10^{-6}(10^{-3})$, $\text{nppwl} = 24$, P1

N	Jacobi	BJ	BGS	BSGS	SGS
5	35	15	10	9	9
10	72	22	14	13	14
20	150	34	22	23	23
40	335	58	38	45	42

Table 4: Volumic preconditioner, homogeneous waveguide with zeroth-order ABC, $k = 20\pi$, $\delta = 4h$, $\text{TOL}=10^{-6}$, $\text{nppwl} = 24$, P1

N	Jacobi	BJ	BGS	BSGS	SGS
5	38	10	6	5	6
10	78	11	7	5	6
20	162	17	9	7	9
40	304	23	12	10	12

Table 5: Volumic preconditioner with PML interface conditions, homogeneous waveguide, $k = 20\pi$, $\delta = 4h$, $\text{TOL}=10^{-6}$, $\text{nppwl} = 24$, P1

δ	Jacobi	BJ	BGS	BSGS	SGS
2	136	19	11	10	10
4	135	18	10	9	9
8	137	17	9	8	9
16	149	21	11	11	11

Table 6: Volumic preconditioner, homogeneous waveguide with zeroth-order ABC, $k = 20$, δ varies, $\text{TOL}=10^{-6}$, $\text{nppwl} = 24$, P1

5.3 Open cavity test

Same as before, the domain is rectangular with an homogeneous medium and its length increases with the number of subdomains. The open cavity test is challenging due to the homogeneous Dirichlet conditions imposed on three sides (cf. black lines in Figure 5). In addition, we perform an excitation on the left side. The Dirichlet conditions create rebounds leading to an increase in the number of reflections, this phenomenon is exacerbated for high-frequency regimes. The global problem can be written as

$$\begin{cases} (-k^2 - \Delta) u = f & \text{in } \Omega \\ (\partial_{\vec{n}} + Ik) u = g & \text{on } \Gamma \\ u = 0 & \text{on } \partial\Omega \setminus \Gamma \end{cases} \quad (44)$$

where $\Gamma := \{x = 0\} \times [0, 1]$ and $g = \exp^{-ik(x\cos(\theta) + y\sin(\theta))}$, $\theta = \frac{\pi}{8}$. It corresponds to an incident plane wave propagating at an angle $\frac{\pi}{8}$ w.r.t. to the horizontal direction, see Fig. 5. This creates numerous reflections on the lateral boundaries of the open cavity. Here as well, there is roughly a factor two in the iteration counts in favour of the SGS, BGS and BSGS algorithms compared to BJ algorithm.

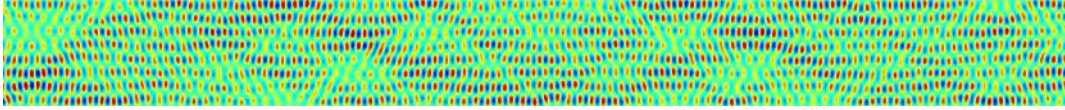


Figure 5: Open cavity solution ($k = 20\pi$)

N	Jacobi	BJ	BGS	BSGS	SGS
5	44 (29)	12 (8)	7 (4)	6 (4)	7 (5)
10	98 (62)	16 (11)	9 (6)	8 (5)	8 (6)
20	202 (142)	21 (14)	11 (8)	10 (8)	11 (8)
40	478 (255)	30 (22)	16 (12)	15 (11)	16 (12)

Table 7: Volumic preconditioner with zeroth-order ABC, open cavity, $k = 20$, $\delta = 4h$, $\text{TOL}=10^{-6}(10^{-3})$, $\text{nppwl} = 24$, P1

N	Jacobi	BJ	BGS	BSGS	SGS
5	49 (33)	12 (8)	6 (4)	6 (4)	7 (4)
10	106 (71)	14 (10)	7 (5)	7 (5)	7 (5)
20	216 (145)	16 (11)	9 (6)	8 (5)	8 (6)
40	427 (260)	19 (12)	10 (7)	10 (6)	10 (7)

Table 8: Volumic preconditioner with PML interface conditions, open cavity, $k = 20$, $\delta = 4h$, $\text{TOL}=10^{-6}(10^{-3})$, $\text{nppwl} = 24$, P1

N	Jacobi	BJ	BGS	BSGS	SGS
5	67 (37)	23 (14)	14 (7)	13 (8)	14 (8)
10	137 (77)	32 (19)	19 (11)	19 (12)	20 (13)
20	298 (152)	50 (31)	29 (21)	31 (23)	32 (23)
40	789 (351)	77 (52)	48 (37)	53 (44)	53 (41)

Table 9: Volumic preconditioner with zeroth-order ABC, open cavity, $k = 20\pi$, $\delta = 4h$, $\text{TOL}=10^{-6}(10^{-3})$, $\text{nppwl} = 24$, P1

5.4 Marmousi

Here we consider the Marmousi benchmark [4]. The velocity profile of the model is depicted in Figure 6. The domain is of size $9.2 \text{ km} \times 3 \text{ km}$. A Neumann boundary condition is imposed at the top boundary, and PMLs are used on the other three boundaries. The source is located at the top. The problem is discretized with P2 elements on a regular mesh with 8 points per wavelength for the reference wavelength corresponding to $c = 2$. Figure 7 shows the real part of the acoustic field at 100

N	Jacobi	BJ	BGS	BSGS	SGS
5	72 (42)	18 (11)	9 (6)	9 (5)	10 (6)
10	153 (84)	19 (11)	10 (6)	9 (5)	11 (6)
20	321 (162)	26 (16)	14 (9)	12 (7)	14 (9)
40	638 (313)	34 (21)	18 (11)	16 (10)	19 (11)

Table 10: Volumic preconditioner with PML interface conditions, open cavity, $k = 20\pi$, $\delta = 4h$, $\text{TOL}=10^{-6}(10^{-3})$, $\text{nppwl} = 24$, P1

Hz frequency. We present results for 25 Hz, 50 Hz and 100 Hz frequencies using the volumic preconditioner. The number of degrees of freedom grows from 1.1 million for $f = 25$ Hz to 4.5 million for $f = 50$ Hz and 17.8 million for $f = 100$ Hz. Table 11 shows numerical results for zeroth order ABC interface conditions and Table 12 for PML. Interestingly we see that for ABC results are quite bad with high iteration counts and dependence on both the number of subdomains and the wave number. Whereas with PML interface conditions, both the iteration counts and their behaviour with respect to the number of subdomains and wavenumber is similar to the previous test cases on waveguides and open cavity. This very bad behaviour of zeroth order ABC at higher frequency and subdomain count can be explained by the variability of the coefficients.

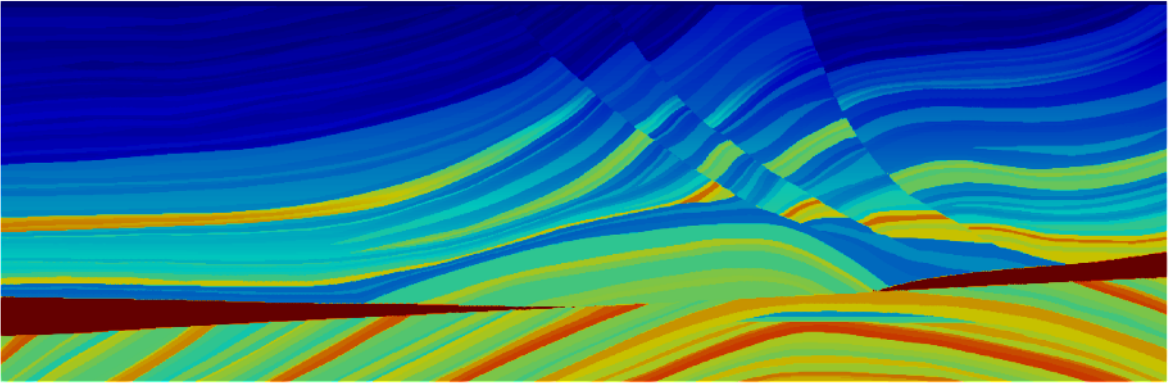


Figure 6: Velocity model of the Marmousi test case

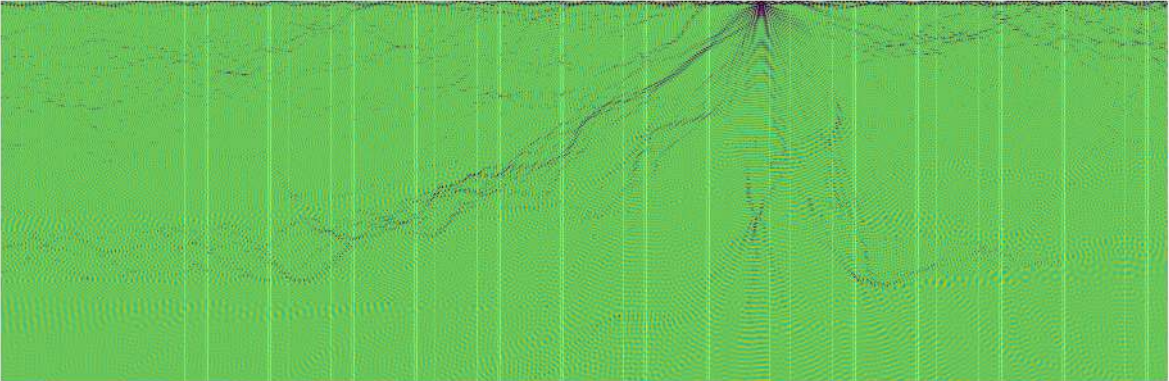


Figure 7: Real part of the solution for $f = 100$ Hz for the Marmousi test case

5.5 3D Overthrust

Here we consider the 3D Overthrust acoustic benchmark. The velocity profile of the model is depicted in Figure 6. The domain is of size $20 \text{ km} \times 20 \text{ km} \times 4.65 \text{ km}$. A Neumann boundary condition is imposed at the top boundary, and PMLs are used on the other five boundaries. The source is located at the top, at $(2.5 \text{ km}, 2.5 \text{ km})$. The problem is discretized with P1 elements on a regular mesh with 10 points per wavelength for the reference wavelength corresponding to $c = 2$. Figure 9 shows the real part of the acoustic field at 4 Hz frequency.

Tables 13 and 14 present results for 1 Hz, 2 Hz and 4 Hz frequencies using the volumic preconditioner with zeroth order ABC and PML interface conditions. The number of degrees of freedom grows from 0.33 million for $f = 1$ Hz to 16.2 million

	25 Hz					50 Hz				
N	Jacobi	BJ	BGS	BSGS	SGS	Jacobi	BJ	BGS	BSGS	SGS
3	22 (12)	11 (6)	10 (5)	7 (4)	7 (4)					
7	46 (25)	17 (9)	13 (7)	10 (6)	11 (6)	49 (25)	19 (9)	16 (8)	13 (8)	14 (8)
14	94 (51)	25 (14)	18 (10)	19 (12)	20(12)	98 (51)	28 (15)	22 (12)	21 (13)	21 (11)
28	185 (100)	41 (24)	27 (16)	30 (20)	30 (19)	195 (101)	49 (26)	38 (21)	60 (48)	47 (32)
56	382 (220)	98 (69)	65 (47)	101 (91)	70 (49)	426 (222)	123(81)	90 (61)	X (192)	143 (116)
112						1505 (690)	X (324)	X (X)	X (X)	X (X)

	100 Hz				
N	Jacobi	BJ	BGS	BSGS	SGS
56	476 (216)	147 (87)	120 (77)	X (X)	208 (171)
112	1701 (691)	X (X)	X (375)	X (X)	X (X)
224	X (X)	X (X)	X (X)	X (X)	X (X)

Table 11: Volumic preconditioner with zeroth-order ABC, Marmousi test case, $\delta = 4h$, $\text{TOL}=10^{-6}(10^{-3})$, $\text{nppwl} = 8$, P2. X means that the algorithm did not reach the convergence criterion in 2000 iterations for Jacobi, or in 400 iterations for the sweeping methods.

	25 Hz					50 Hz				
N	Jacobi	BJ	BGS	BSGS	SGS	Jacobi	BJ	BGS	BSGS	SGS
3	14 (8)	7 (4)	6 (3)	4 (2)	5 (3)					
7	33 (17)	10 (5)	8 (4)	5 (3)	6 (4)	34 (18)	10 (6)	9 (4)	6 (3)	6 (4)
14	64 (35)	11 (6)	8 (5)	6 (4)	7 (4)	69 (35)	12 (6)	10 (5)	7 (4)	8 (4)
28	126 (66)	13 (7)	9 (5)	7 (4)	8 (4)	133 (66)	14 (7)	11 (6)	7 (4)	8 (5)
56	247 (124)	18 (11)	12 (7)	11 (6)	12 (6)	260 (125)	18 (9)	13 (7)	10 (5)	11 (5)
112						531 (240)	32 (17)	22 (12)	20 (11)	21 (11)

	100 Hz				
N	Jacobi	BJ	BGS	BSGS	SGS
56	242 (103)	18 (9)	14 (6)	10 (5)	11 (5)
112	483 (198)	29 (13)	20 (9)	15 (7)	18 (8)
224	1069 (417)	66 (32)	42 (20)	62 (44)	38 (18)

Table 12: Volumic preconditioner with PML interface conditions, Marmousi test case, $\delta = 4h$, $\text{TOL}=10^{-6}(10^{-3})$, $\text{nppwl} = 8$, P2

for $f = 4$ Hz. We observe similar trends as for the Marmousi test case, although the zeroth order ABC case is not so bad, as the frequency and number of wavelengths in one direction is not as high for this 3D test case. Moreover, we can see that the iteration count is much more dependent on the number of subdomains than on the frequency.

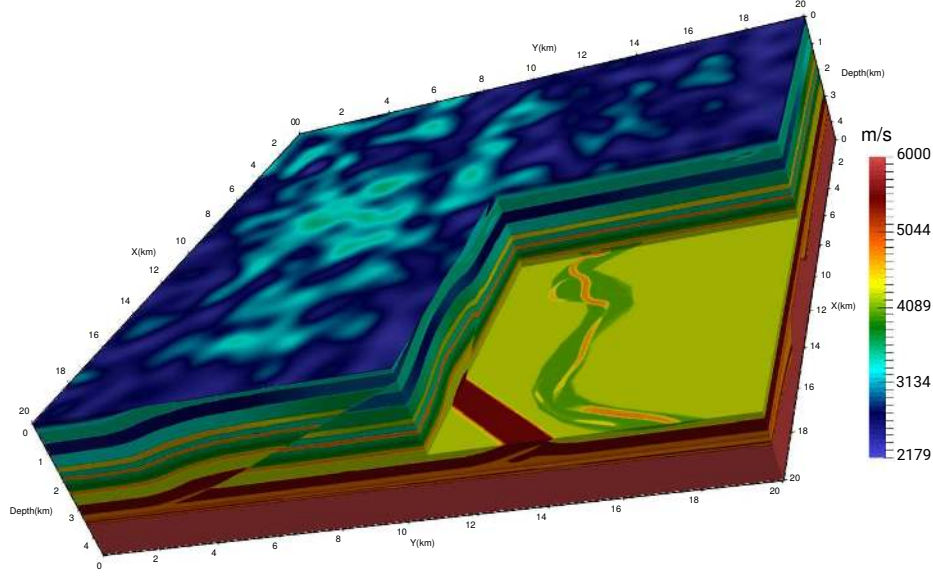


Figure 8: Velocity model of the 3D Overthrust benchmark

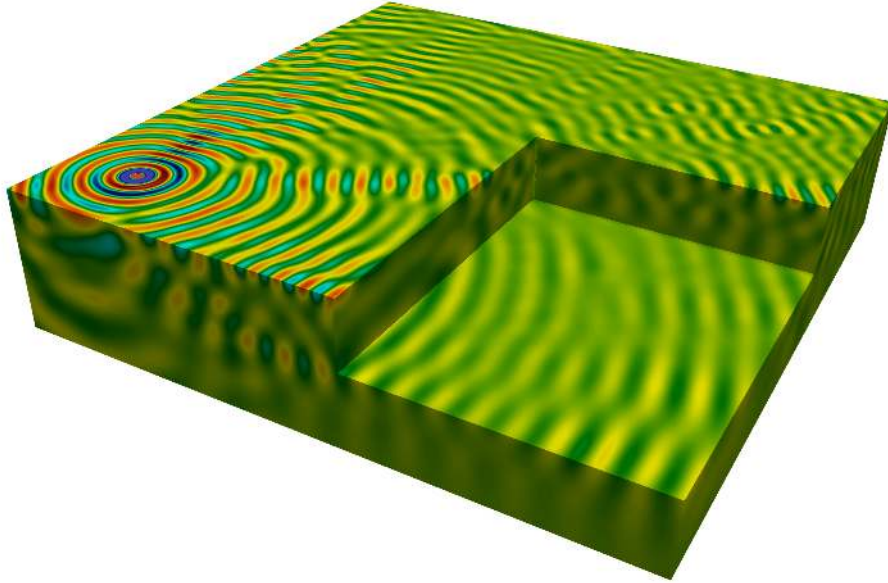


Figure 9: Real part of the solution for $f = 4$ Hz for the 3D Overthrust benchmark

6 Conclusion

We have introduced a unified framework for several sweeping algorithms which eases their derivation and theoretical studies, see Table 1. We have focused on two double sweep methods: SGS of [23, 31] and BJ of [27, 30]. We conclude both from their theoretical convergence rates and from our numerical results that the algorithm SGS has a better behavior than the BJ algorithm. The numerical experiments also illustrate the superiority of PML as interface conditions compared to zeroth order ABCs, especially in the variable coefficient case. Although these algorithms are more easily derived in their substructured form, they lend themselves to volumic variants that allow for inexact solves in the subdomains, see § 4.6. In this form, these methods should have then a behavior similar to the sweeping algorithm of [13].

	1 Hz					2 Hz				
N	Jacobi	BJ	BGS	BSGS	SGS	Jacobi	BJ	BGS	BSGS	SGS
3	14 (7)	7 (4)	6 (3)	4 (2)	4 (2)					
7	32 (17)	9 (5)	7 (4)	5 (3)	6 (3)	34 (17)	10 (5)	8 (4)	5 (3)	6 (3)
14	94 (45)	18 (9)	10 (5)	9 (5)	9 (5)	68 (34)	15 (8)	10 (5)	8 (4)	9 (4)
28						163 (80)	30 (16)	16 (9)	17 (10)	14 (8)

	4 Hz				
N	Jacobi	BJ	BGS	BSGS	SGS
28	162 (80)	32 (17)	19 (10)	24 (15)	17 (9)
56	365 (187)	87 (48)	42 (22)	168 (133)	36 (20)

Table 13: Volumic preconditioner with zeroth-order ABC, Overthrust test case, $\delta = 4h$, $TOL=10^{-6}(10^{-3})$, $nppwl = 10$, P1

	1 Hz					2 Hz				
N	Jacobi	BJ	BGS	BSGS	SGS	Jacobi	BJ	BGS	BSGS	SGS
3	11 (7)	5 (3)	4 (2)	3 (2)	3 (2)					
7	28 (15)	7 (4)	5 (3)	4 (2)	5 (3)	30 (16)	8 (4)	6 (3)	4 (2)	5 (3)
14	59 (31)	9 (5)	6 (4)	5 (3)	6 (3)	63 (31)	11 (6)	7 (4)	6 (3)	7 (3)
28						123 (61)	16 (8)	10 (5)	7 (4)	9 (5)

	4 Hz				
N	Jacobi	BJ	BGS	BSGS	SGS
28	121 (63)	18 (9)	11 (6)	10 (6)	11 (6)
56	224 (122)	28 (15)	18 (9)	19 (12)	16 (9)

Table 14: Volumic preconditioner with PML interface conditions, Overthrust test case, $\delta = 4h$, $TOL=10^{-6}(10^{-3})$, $nppwl = 10$, P1

Also, an intrinsic problem with double sweep methods is that due to the sequentiality of the algorithm, subdomains are idle most of the time. To be more precise, in SGS, only one process is active at a time whereas in BJ, two processes are active at a time (one for the left sweep and one for the right sweep performed concurrently). A solution to overcome this is to introduce a pipelining technique that can be applied to multiple right-hand sides problems to improve parallelism and achieve significant speed-ups, see [27, 30].

We have considered stripwise decompositions and it would be interesting to see if the present analysis can be extended to L-sweeps preconditioners (see e.g. [29, 7] and references therein) used for checkerboard type decompositions.

Acknowledgment

This work was granted access to the HPC resources of OCCIGEN@CINES under the allocation 2021-067730 granted by GENCI. The first author was partially supported by TOTAL Centre Scientifique et Technique Jean Féger.

References

- [1] Xavier Antoine, Yassine Boubendir, and Christophe Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. *Journal of Computational Physics*, 231(2):262–280, 2012.
- [2] Alvin Bayliss and Eli Turkel. Radiation boundary conditions for wave-like equations. *Comm. Pure Appl. Math.*, 33(6):707–725, 1980.
- [3] Alfredo Bermúdez, L Hervella-Nieto, A Prieto, and R Rodríguez. An exact bounded pml for the helmholtz equation. *Comptes Rendus Mathématique*, 339(11):803–808, 2004.
- [4] A Brougois, Marielle Bourget, Patriek Lailly, Michel Poulet, Patrice Ricarte, and Roelof Versteeg. Marmousi, model and data. In *EAEG workshop-practical aspects of seismic data inversion*, pages cp–108. European Association of Geoscientists & Engineers, 1990.
- [5] Zhiming Chen and Xueshuang Xiang. A source transfer domain decomposition method for helmholtz equations in unbounded domain. *SIAM Journal on Numerical Analysis*, 51(4):2331–2356, 2013.

- [6] W. C. Chew and W. H. Weedon. A 3d perfectly matched medium from modified maxwell's equations with stretched coordinates. *IEEE Trans. Microwave Opt. Technol. Lett.*, 7:599–604, 1994.
- [7] Ruiyang Dai, Axel Modave, Jean-François Remacle, and Christophe Geuzaine. Multidirectional sweeping preconditioners with non-overlapping checkerboard domain decomposition for helmholtz problems. *Journal of Computational Physics*, 453:110887, 2022.
- [8] Bruno Després. Décomposition de domaine et problème de Helmholtz. *C.R. Acad. Sci. Paris*, 1(6):313–316, 1990.
- [9] Bruno Després. Domain decomposition method and the helmholtz problem. In *Mathematical and numerical aspects of wave propagation phenomena (Strasbourg, 1991)*, pages 44–52, Philadelphia, PA, 1991. SIAM.
- [10] Bruno Després, Patrick Joly, and Jean E. Roberts. A domain decomposition method for the harmonic Maxwell equations. In *Iterative methods in linear algebra (Brussels, 1991)*, pages 475–484, Amsterdam, 1992. North-Holland.
- [11] Bjorn Engquist and Andrew Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, 31(139):629–651, 1977.
- [12] Björn Engquist and Lexing Ying. Sweeping preconditioner for the helmholtz equation: Hierarchical matrix representation. *Communications on Pure and Applied Mathematics*, 64(5):697–735, 2011.
- [13] Björn Engquist and Lexing Ying. Sweeping preconditioner for the helmholtz equation: Moving perfectly matched layers. *Multiscale Modeling & Simulation*, 9(2):686–710, 2011.
- [14] Martin J. Gander, Frédéric Magoulès, and Frédéric Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24(1):38–60, 2002.
- [15] Martin J Gander and Hui Zhang. A class of iterative solvers for the helmholtz equation: Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized schwarz methods. *Siam Review*, 61(1):3–76, 2019.
- [16] Shihua Gong, Ivan G Graham, and Euan A Spence. Convergence of restricted additive schwarz with impedance transmission conditions for discretised helmholtz problems. *arXiv preprint arXiv:2110.14495*, 2021.
- [17] Thomas Hagstrom, R. P. Tewarson, and Aron Jazcilevich. Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems. *Appl. Math. Lett.*, 1(3), 1988.
- [18] F. Hecht. New development in Freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.
- [19] J.P.Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. of Comp.Phys.*, 114:185–200, 1994.
- [20] Pierre-Louis Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1990. SIAM.
- [21] M. Lothaire. *Combinatorics on words*, volume 17. Cambridge university press, 1997.
- [22] Frédéric Nataf. Absorbing boundary conditions and perfectly matched layers in wave propagation problems. *Direct and inverse problems in wave propagation and applications*, 14:219–231, 2013.
- [23] Frédéric Nataf and Francis Nier. Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. *Numerische Mathematik*, 75(3):357–77, 1997.
- [24] Frédéric Nataf, Francois Rogier, and Eric de Sturler. Optimal interface conditions for domain decomposition methods. Technical Report 301, CMAP (Ecole Polytechnique), <https://hal.archives-ouvertes.fr/hal-02194208>, 1994.
- [25] Youssef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7:856–869, 1986.
- [26] H. A. Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, 1870.
- [27] Christiaan C. Stolk. A rapidly converging domain decomposition method for the Helmholtz equation. *Journal of Computational Physics*, 241(0):240 – 252, 2013.

- [28] Christiaan C. Stolk. An improved sweeping domain decomposition preconditioner for the helmholtz equation. *Advances in Computational Mathematics*, 43(1):45–76, 2017.
- [29] Matthias Taus, Leonardo Zepeda-Núñez, Russell J. Hewett, and Laurent Demanet. L-sweeps: A scalable, parallel preconditioner for the high-frequency helmholtz equation. *Journal of Computational Physics*, 420:109706, 2020.
- [30] Alexandre Vion and Christophe Geuzaine. Double sweep preconditioner for optimized schwarz methods applied to the helmholtz problem. *Journal of Computational Physics*, 266:171–190, 2014.
- [31] Alexandre Vion and Christophe Geuzaine. Improved sweeping preconditioners for domain decomposition algorithms applied to time-harmonic helmholtz and maxwell problems. *ESAIM: Proceedings and Surveys*, 61:93–111, 2018.
- [32] Leonardo Zepeda-Núñez and Laurent Demanet. The method of polarized traces for the 2d helmholtz equation. *Journal of Computational Physics*, 308:347–388, 2016.