

IMPROVED SWEEPING PRECONDITIONERS FOR DOMAIN DECOMPOSITION ALGORITHMS APPLIED TO TIME-HARMONIC HELMHOLTZ AND MAXWELL PROBLEMS^{*,**}ALEXANDRE VION AND CHRISTOPHE GEUZAINÉ¹

Abstract. Sweeping-type algorithms have recently gained a lot of interest for the solution of high-frequency time-harmonic wave problems, in particular when used in combination with perfectly matched layers. However, an inherent problem with sweeping approaches is the sequential nature of the process, which makes them inadequate for efficient implementation on parallel computers. We propose several improvements to the double-sweep preconditioner originally presented in [18], which uses sweeping as a matrix-free preconditioner for a Schwarz domain decomposition method. Similarly, the improved preconditioners are based on approximations of the inverse of the Schwarz iteration operator: the general methodology is to apply well-known algebraic techniques to the operator seen as a matrix, which in turn is processed to obtain equivalent matrix-free routines that we use as preconditioners. A notable feature of the new variants is the introduction of partial sweeps that can be performed concurrently in order to make a better usage of the resources. As these modifications still leave some unexploited computational power, we also propose to combine them with right-hand side pipelining to further improve parallelism and achieve significant speed-ups. Examples are presented on high-frequency Helmholtz and Maxwell problems, in two and three dimensions, to demonstrate the properties of our improvements on parallel computer architectures.

1. INTRODUCTION

Solving time-harmonic wave equations using finite element-type methods is a notoriously difficult problem, especially in the high-frequency regime where the number of unknowns becomes so large that the direct solution of the resulting linear system is computationally intractable. Iterative solvers based on a domain decomposition method and a sweeping process over the subdomains appear as an interesting approach: these techniques use direct solutions on slices of the domain, and propagate information across the slices using some transmission condition, among which perfectly matched layers have recently deserved special attention [6, 7, 12, 14, 18]. The sweeping process, which is linked to the symbolic block factorization of the problem, is inherently sequential: each slice must be solved before the next one can be considered. So, although the number of iterations of the iterative algorithm can be shown to be independent of frequency under some conditions, sweeping algorithms do not scale on parallel computer architectures, as only the (direct) solution in each slice can be parallelized.

* This work was funded in part through the ARC grant for Concerted Research Actions (ARC WAVES 15/19-03), financed by the Wallonia-Brussels Federation, as well as through the Belgian Science Policy (PAI grant P7/02).

** Computational resources have been provided by CÉCI, funded by F.R.S.-FNRS (Fonds de la Recherche Scientifique) under grant n°2.5020.11, and the Tier-1 supercomputer of the Fédération Wallonie-Bruzelles, funded by the Walloon Region under grant n° 1117545.

¹ Université de Liège, Institut Montefiore B28, B-4000 Liège, Belgium, cgeuzaine@ulg.ac.be.

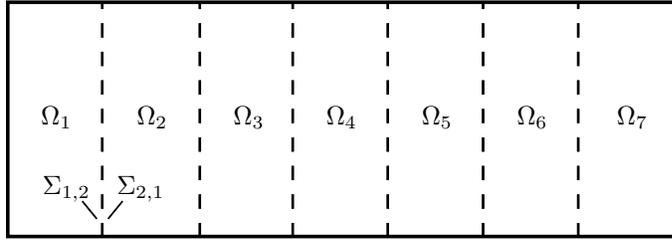


FIGURE 1. Layered decomposition of the domain, without overlap.

The double-sweep preconditioner originally proposed in [18] introduced sweeping naturally as a matrix-free preconditioner for a non-overlapping Schwarz domain decomposition method. Recasting the Schwarz domain decomposition algorithm as the solution of a linear system, the main idea was to derive an explicit, approximate inverse of the Schwarz iteration matrix in the case of a simple one-dimensional, layer-like decomposition of the computational domain. This explicit inverse could be written recursively as the computation of two concurrent sweeps across the subdomains, one in the forward direction and one backwards. While other sweeping approaches directly precondition or solve the discrete wave equation problem, this approach preconditions the iteration operator of the domain decomposition solver, leading to very small problem sizes (the unknowns are located only on the interfaces between the subdomains). Provided that accurate approximations of the Dirichlet-to-Neumann (DtN) operator exist on the boundaries between the subdomains (e.g. using high-order local transmission conditions [3, 4, 8, 13] or perfectly matched layers [1, 2, 11]), an iteration count independent of the number of subdomains and frequency is observed in practical numerical simulations. While this approach possesses several advantages compared to competing sweeping techniques (notably its matrix-free nature and the ease with which it can be integrated in existing simulation codes), each sweep is still a sequential process [12, 17].

In this paper we investigate several variants of this preconditioner, by considering different approximations of the inverse of the Schwarz iteration operator. Some of the new variants can be linked to well-known algebraic techniques (Block-Jacobi, Gauss-Seidel); thanks to the introduction of cuts in the sweeping process, multiple sub-sweeps can be performed concurrently and thus lead to enhanced scalability on parallel computer architectures.

The paper is organized as follows. Section 2 recalls the classical non-overlapping Schwarz algorithm for a layered decomposition of the computational domain, and recasts it as the solution of a linear system of equations. In Section 3 we take advantage of the matrix representation of the Schwarz iteration operator to design a family of preconditioners, the original double-sweep preconditioner from [18] being a particular case. An extension with shorter, concurrent sweeps is proposed, as well as two Gauss-Seidel variants. Section 4 then presents a pipelining strategy to further improve parallelism when the new sweeping preconditioners are used to solve problems with multiple right-hand-sides. Finally, numerical results on large acoustic (Helmholtz) and electromagnetic (Maxwell) problems are presented in Section 5.

2. NON-OVERLAPPING SCHWARZ ALGORITHM

Let us begin by describing the well-known non-overlapping Schwarz algorithm using operator notations. This will be useful for describing the overall algorithm as the application of an iteration operator, represented by a matrix, without detailing the underlying partial differential equations or the discretization scheme. The sweeping preconditioners presented in Section 3 will be derived directly using this operator form. The numerical results presented in Section 5 will use these preconditioners in combination with a finite element discretization of both Helmholtz (scalar) and Maxwell's (vector) equations: for a detailed presentation of the finite element formulations and their implementation in the context of Schwarz domain decomposition methods, see [15].

2.1. Layered Decomposition of the Domain

We consider a “layered”, non-overlapping decomposition of the domain Ω with boundary Γ (and exterior unit normal \mathbf{n}), where each subdomain Ω_i , $i = 1, \dots, N$, has at most two strictly non-adjacent neighbors and the first and last subdomains are unconnected, as illustrated on Figure 1. Such one-dimensional decompositions are naturally well suited for the sweeping methods that we are considering. (The case of “cyclic” decompositions where the first and last domains have a common boundary can be treated in a similar way.) For every $i = 1, \dots, N$, we define $\Gamma_i = \Gamma \cap \partial\Omega_i$; and for $j = 1, \dots, N, j \neq i$, we introduce the artificial interface (or “transmission boundary”) $\Sigma_{i,j} = \Sigma_{j,i} = \overline{\partial\Omega_i} \cap \overline{\partial\Omega_j}$. For the considered layered decompositions, $\Sigma_{i,j} \neq \emptyset$ if $|i - j| = 1$. We also denote by \mathbf{n}_i the unit exterior normal to the subdomain Ω_i .

2.2. Algorithm in Operator Form

A generic time-harmonic wave propagation problem with sources in the domain Ω and homogeneous boundary conditions on Γ can be written in the form

$$\begin{aligned} \mathcal{L}u &= f & \text{in } \Omega, \\ \mathcal{B}u &= 0 & \text{on } \Gamma, \end{aligned} \quad (1)$$

where \mathcal{L} and \mathcal{B} are linear operators. For example, for an acoustic wave problem with a Sommerfeld radiation condition (the unknown scalar field u denoting the pressure), one would have $\mathcal{L} := (\Delta + k^2)$ (with k the acoustic wavenumber) and $\mathcal{B} := (\partial_{\mathbf{n}} - ik)$. For an electromagnetic wave problem with a Silver-Müller absorbing boundary condition (the unknown vector field u now denoting the electric field), one would have $\mathcal{L} := (\mathbf{curl} \mathbf{curl} - k^2)$ (with k the electromagnetic wavenumber) and $\mathcal{B} := (\gamma_{\mathbf{n}}^t \mathbf{curl} + ik\gamma_{\mathbf{n}}^T)$, with $\gamma_{\mathbf{n}}^t$ and $\gamma_{\mathbf{n}}^T$ respectively the tangential and tangential component trace operators ($\gamma_{\mathbf{n}}^t : \mathbf{v} \mapsto \mathbf{n} \times \mathbf{v}$ and $\gamma_{\mathbf{n}}^T : \mathbf{v} \mapsto \mathbf{n} \times (\mathbf{v} \times \mathbf{n})$). Different sources and more general boundary conditions can be considered without difficulty.

The additive Schwarz domain decomposition algorithm can be described as follows at iteration $k+1$, assuming e.g. $g_{i,j}^{(0)} = 0$. For $i = 1, \dots, N$, compute $u_i^{(k+1)}$ solution to

$$\begin{aligned} \mathcal{L}u_i^{(k+1)} &= f & \text{in } \Omega_i, \\ \mathcal{B}u_i^{(k+1)} &= 0 & \text{on } \Gamma_i, \\ \mathcal{B}_{i,j}u_i^{(k+1)} &= g_{i,j}^{(k)} & \text{on } \Sigma_{i,j}. \end{aligned} \quad (2)$$

Then for $i, j = 1, \dots, N$ such that $|i - j| = 1$, update the interface unknowns according to:

$$g_{j,i}^{(k+1)} = \mathcal{B}_{j,i}u_i^{(k+1)} \quad \text{on } \Sigma_{j,i}. \quad (3)$$

Detailed expressions for the operators $\mathcal{B}_{i,j}$ in the case of time-harmonic acoustic and electromagnetic wave problems can be found in [15]: they are respectively of the general form $\mathcal{B}_{i,j} = \partial_{\mathbf{n}_i} + \mathcal{S}$ for acoustics, and $\mathcal{B}_{i,j} = (\gamma_{\mathbf{n}_i}^t \mathbf{curl} + \mathcal{S})$ for electromagnetics, where \mathcal{S} is called the transmission operator. The simplest, lowest order transmission operator for acoustics is $\mathcal{S} := -ik$, and $\mathcal{S} := ik\gamma_{\mathbf{n}_i}^T = ik\gamma_{\mathbf{n}_j}^T$ for electromagnetics. Physically, the nonhomogeneous boundary conditions on the artificial interfaces $\Sigma_{i,j}$ can be interpreted as (generalized) impedance boundary conditions.

Thanks to the general form of the operators $\mathcal{B}_{i,j}$, in the case of non-overlapping decompositions the update relation (3) can be rewritten as

$$g_{j,i}^{(k+1)} = \mathcal{B}_{j,i}u_i^{(k+1)} = -g_{i,j}^{(k)} + 2\mathcal{S}u_i^{(k+1)}, \quad (4)$$

since $\mathbf{n}_i = -\mathbf{n}_j$ on $\Sigma_{i,j} = \Sigma_{j,i}$.

Considering the full vector of unknowns $g = [g_{1,2}, g_{2,1}, g_{2,3}, \dots]^T$ made of all the interface unknowns $g_{i,j}$, one step of the above algorithm can be summarized as:

$$g^{(k+1)} = \mathcal{A}g^{(k)} + b, \quad (5)$$

for some right hand side b . Iteration (5) is a fixed-point iteration, the solution of which solves the linear system that we will call the Schwarz problem:

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

Letting aside the choice of a discretization method, the full algorithm can thus be decomposed into four steps (see [15] for details):

1. Assemble and factorize the system matrix for each one of the subproblems (2);
2. Compute the right-hand side of the Schwarz problem (6), that contains the contributions of the (physical) sources;
3. Solve the Schwarz problem (6) for the interface unknowns g ;
4. Compute the solution u_i in each of the subdomains Ω_i , based on g .

Steps 2–4 constitute the “online” part of the algorithm that must be repeated for any new right-hand side (step 1 can be performed once, offline). From a parallel computing perspective, the most challenging step is 3: all the others are independent and can be carried out simultaneously without communication. In practice (6) is solved using an iterative Krylov subspace method such as GMRES or ORTHODIR, which only requires the application of the operator \mathcal{F} on a given iterate. This application involves the solution of the subproblems, and is thus usually made “matrix-free”, i.e. without explicitly constructing the associated matrix coefficients.

As will be explained in the next section, however, one can provide an abstract matrix representation by associating transfer operators to the subproblem solves, that we will use later on to design a family of preconditioners for that system.

2.3. Matrix Form of the Iteration Operator

For our layered decomposition of the domain, the unknown vector g of the Schwarz algorithm contains a total of $M = 2(N - 1)$ unknown fields, which is thus the size of the (continuous) Schwarz problem. We introduce the forward and backward transfer operators \mathcal{B}_i^f and \mathcal{B}_i^b across Ω_i , defined by:

$$\begin{aligned} \mathcal{B}_i^f : \quad g_{i,i-1} &\longmapsto -2\mathcal{S}u_i(g_{i,i-1}, 0)|_{\Sigma_{i,i+1}} := \mathcal{B}_i^f g_{i,i-1}, \\ \mathcal{B}_i^b : \quad g_{i,i+1} &\longmapsto -2\mathcal{S}u_i(0, g_{i,i+1})|_{\Sigma_{i,i-1}} := \mathcal{B}_i^b g_{i,i+1}, \end{aligned} \quad (7)$$

where $u_i(g_{i,i-1}, g_{i,i+1})|_{\Sigma}$ refers to the restriction on Σ of the solution to the subproblem $\mathcal{L}u_i = f$ defined by (2). Notice that the forward and backward transfer operators only involve the solution of subproblems with a nonhomogeneous impedance boundary condition on one side only (left or right), since either $g_{i,i-1}$ or $g_{i,i+1}$ is set to 0. We also define the self-coupling operators \mathcal{E}_i^f and \mathcal{E}_i^b , defined from an interface to itself:

$$\begin{aligned} \mathcal{E}_i^f : \quad g_{i,i-1} &\longmapsto g_{i,i-1} - 2\mathcal{S}u_i(g_{i,i-1}, 0)|_{\Sigma_{i,i-1}} = \mathcal{E}_i^f g_{i,i-1}; \\ \mathcal{E}_i^b : \quad g_{i,i+1} &\longmapsto g_{i,i+1} - 2\mathcal{S}u_i(0, g_{i,i+1})|_{\Sigma_{i,i+1}} = \mathcal{E}_i^b g_{i,i+1}. \end{aligned} \quad (8)$$

These operators correspond to the contribution on an interface of the part of a wave that travels through the domain and that is reflected back to its interface of origin. These operators are illustrated on Figure 2.

We now take advantage of this recurrence property to write the matrix-vector product $g' = \mathcal{F}_A^{-1}r$ (in the context of right-preconditioned linear solvers, r denotes the residual) as follows:

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

where any sum $\sum_{p=a}^b$ for which $a > b$ should be ignored. Factoring these expressions, we rewrite them as:

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

where the operators \mathcal{B}_i^f with $i \leq 1$ and \mathcal{B}_i^b with $i \geq N$ are not defined and should be ignored. This finally gives the double recurrence relation:

$$\begin{aligned} g'_{2,1} &= r_{2,1}; \\ g'_{i+1,i} &= r_{i+1,i} - \mathcal{B}_i^f g'_{i,i-1}, \quad i = 2, \dots, N-1; \\ g'_{N-1,N} &= r_{N-1,N}; \\ g'_{i-1,i} &= r_{i-1,i} - \mathcal{B}_i^b g'_{i,i+1}, \quad i = N-1, \dots, 2. \end{aligned} \tag{13}$$

The first relation describes a forward sweep: we start from the first artificial interface $\Sigma_{1,2}$ and propagate the information by solving a problem at each step to move to the next interface and incorporate its contribution. The other relation describes the same procedure in the backward direction; because these sequences are independent of each other (they act on distinct subsets of the unknowns), they can be done in parallel. We note that the extreme subproblems (the first and the N -th) are not solved in any of the sequences, so each sweep has an actual length of $N-2$ steps.

Another convenient way of rewriting relations (13) is to use multi-indices I_k that are dense collections of sorted subdomain indices belonging to the k -th group of $N_k \geq 2$ subdomains. (So far we worked with a single group made of all subdomains (I_1), but we will later subdivide them.) We also define I'_k , where the sequence of indices is reversed. We now define the sweeps of length $l > 0$ in the k -th group as:

$$\begin{aligned} S_k^f(l) &= \sum_{p=I_k(2)}^{I_k(l+1)} \left(\prod_{q=I_k(l+1)}^p -\mathcal{B}_q^f \right) r_{p,p-1}; \\ S_k^b(l) &= \sum_{p=I_k(2)}^{I'_k(l+1)} \left(\prod_{q=I_k(l+1)}^p -\mathcal{B}_q^b \right) r_{p,p+1}, \end{aligned}$$

with $S_k^{\{f,b\}}(0) = 0$. Then we have the induction property for $1 \leq l \leq N_k - 2$:

$$\begin{aligned} S_k^f(l) &= -\mathcal{B}_{I_k(l+1)}^f (r_{I_k(l+1),I_k(l)} + S_k^f(l-1)); \\ S_k^b(l) &= -\mathcal{B}_{I'_k(l+1)}^b (r_{I'_k(l+1),I'_k(l)} + S_k^b(l-1)), \end{aligned}$$

and the recurrence relations rewrite simply as:

$$\begin{aligned} g'_{i,i-1} &= r_{i,i-1} + S_1^f(i - I_1(2)), & i &= I_1(2), \dots, I_1'(1) = 2, \dots, N; \\ g'_{i,i+1} &= r_{i,i+1} + S_1^b(I_1'(2) - i), & i &= I_1'(2), \dots, I_1(1) = N - 1, \dots, 1. \end{aligned} \tag{14}$$

In our numerical experiments with the double sweep (see [16, 18]), we have observed that, provided that the approximation of the DtN map used in the transmission condition is sufficiently accurate, the preconditioned Schwarz algorithm has the desirable properties of having a convergence rate virtually independent of the wavenumber k and number of subdomains N , even in the case of a heterogeneous medium. For complex heterogeneous media, the most accurate approximations of the DtN were obtained using Perfectly Matched Layers [16, 18]. A detailed description of the various DtN approximations and their implementation in a finite element context, for both acoustic and electromagnetic problems, is provided in [15].

Examining the double sweep algorithm, it is important to note that it only involves subproblem solutions that are the same as the one performed in the main iteration, albeit with different right-hand-sides. In particular this means that no specific preprocessing is required to build the standard double sweep preconditioner. In total, the standard double sweep algorithm requires $N + 2(N - 2) = 3N - 4$ subproblem solutions per iteration.

It is interesting to note that an equivalent algorithm can be derived by combining the application of the preconditioner and the Schwarz operator: for right-preconditioning it gives

$$\mathcal{F}_c = \mathcal{F}\mathcal{F}_A^{-1}.$$

We denote the resulting algorithm as the combined double sweep preconditioner, which amounts to solve an unpreconditioned problem:

$$\mathcal{F}_c g = \mathcal{F}_A^{-1} b.$$

Applying \mathcal{F}_c rather than its two factors allows to bring down the total number of subproblem solves to $2N - 2$ [18] by skipping redundant subproblem solves. It also has the fully parallel step removed and only performs sweeps, which will let us pipeline several right-hand sides as will be detailed in Section 4. Although an apparent link with multiplicative Schwarz can be seen in this combined sweep procedure, we point out some differences: simple multiplicative Schwarz sequentially solves all problem in an arbitrarily chosen direction, while we do the sweeps in both directions (which is closer to a symmetrical Gauss-Seidel iteration); the way information is exchanged and stored during the sweep is also different.

The main drawback of sweeping preconditioners is the sequential nature of the sweeps. If each CPU has a single subdomain assigned to it, it will remain idle during most of the duration of the sweep, as illustrated on Figure 4 (left). The next section introduces cuts in the sweeps in order to improve parallelism.

3.2. Block-Jacobi Preconditioners: Double Sweep with Cuts

If we introduce one or more cuts in the sweeping process, the resulting partial sweeps can be performed independently and in parallel. This is akin to a block-Jacobi preconditioning strategy, improving parallelism at the cost of a less accurate approximation of the inverse of \mathcal{F}_A .

The matrix corresponding to these reduced sweeps is derived from the full preconditioner matrix, where the transmission operators corresponding to the cut domain is replaced by 0. As a consequence, communication between the two groups of domains is forbidden and the preconditioner matrix has well separated blocks, that correspond to the independent restricted double sweeps. Each block is the inverse of a Schwarz operator \mathcal{F}_{A_i} over the i -th portion of the domain consisting of a few adjacent domains, taken apart from the other portions. For the situation of Figure 3, where portions 1 and 2 have multi-indices $I_1 = \{1, 2, 3, 4\}$ and $I_2 = \{4, 5, 6, 7\}$

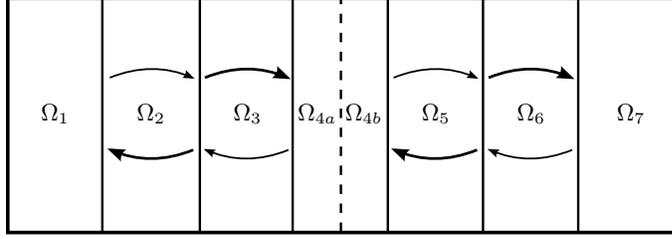


FIGURE 3. Partial sweeps cover non-overlapping groups of domains, separated by the dashed line. The position of the cut inside the domain is not important as the first and last domains are not solved in our sweeps, as shown by the arrows.

respectively, the matrix is:

$$\mathcal{F}_{A_{\text{cut}}}^{-1} = \begin{bmatrix} \begin{bmatrix} \mathcal{I} & 0 & -\mathcal{B}_2^b & 0 & \mathcal{B}_2^b \mathcal{B}_3^b & 0 \\ 0 & \mathcal{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{I} & 0 & -\mathcal{B}_3^b & 0 \\ 0 & -\mathcal{B}_2^f & 0 & \mathcal{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{I} \\ 0 & \mathcal{B}_3^f \mathcal{B}_2^f & 0 & -\mathcal{B}_3^f & 0 & \mathcal{I} \end{bmatrix} & \begin{bmatrix} \mathcal{I} & 0 & -\mathcal{B}_5^b & 0 & \mathcal{B}_5^b \mathcal{B}_6^b & 0 \\ 0 & \mathcal{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{I} & 0 & -\mathcal{B}_6^b & 0 \\ 0 & -\mathcal{B}_5^f & 0 & \mathcal{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{I} & 0 \\ 0 & \mathcal{B}_6^f \mathcal{B}_5^f & 0 & -\mathcal{B}_6^f & 0 & \mathcal{I} \end{bmatrix} \end{bmatrix} \quad (15)$$

$$= \begin{bmatrix} \mathcal{F}_{A_1}^{-1} & \\ & \mathcal{F}_{A_2}^{-1} \end{bmatrix}.$$

This is very much like a block-Jacobi preconditioner for the simplified Schwarz iteration operator. In the k -th group of subdomains, we have the relations:

$$\begin{aligned} g'_{i,i-1} &= r_{i,i-1} + S_k^f(i - I_k(2)), & i &= I_k(2), \dots, I_k'(1) \\ g'_{i,i+1} &= r_{i,i+1} + S_k^b(I_k'(2) - i), & i &= I_k'(2), \dots, I_k(1). \end{aligned} \quad (16)$$

In the ideal case where the exact DtN is used as transmission condition, the preconditioned operator has the following structure:

$$\mathcal{F}_A \mathcal{F}_{A_{\text{cut}}}^{-1} = \begin{bmatrix} \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{I} & 0 & \mathcal{B}_4^b & 0 & -\mathcal{B}_4^b \mathcal{B}_5^b & 0 & \mathcal{B}_4^b \mathcal{B}_5^b \mathcal{B}_6^b & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{B}_4^f \mathcal{B}_3^f \mathcal{B}_2^f & 0 & -\mathcal{B}_4^f \mathcal{B}_3^f & 0 & \mathcal{B}_4^f & 0 & \mathcal{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} \end{bmatrix}, \quad (17)$$

that is relatively close to an identity. Spectral analysis reveals that the eigenvalues are well clustered ($\lambda_{1, \dots, 2N-2} = 1$) but that it is defective, as is the unpreconditioned Schwarz operator. However, the number of missing eigenvectors is $2N_c$ instead of $2N - 4$, if N_c is the number of cuts, which is a strong improvement if $N_c \ll N$. Numerical experiments with this new version of the preconditioner confirm this analysis, with a convergence

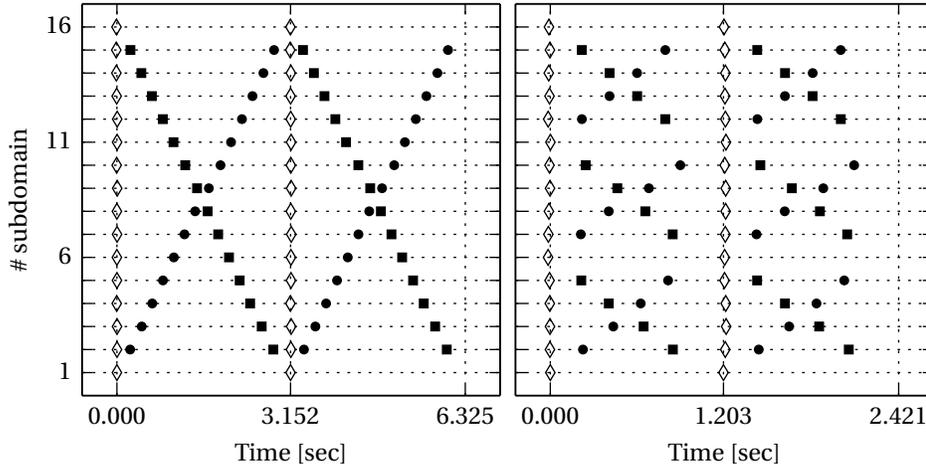


FIGURE 4. Introducing 2 cuts in the double sweep preconditioner (right) enables parallel execution of the partial sweeps, reducing the application time of the preconditioner without cuts (left). The white diamonds indicate solves performed in the iteration operator; the black circles and squares indicate solves in the forward and backward sweeps, respectively (we are thus in the case of left preconditioning). These timelines were obtained for the COBRA test case of Section 5.2, with 16 subdomains and cuts in subdomains 6 and 11.

rate that degrades as the number of cuts increases for a strongly guided problem, and plateaus appearing on the convergence curve. This situation is to be compared with the behavior of the unpreconditioned DDM algorithm, that corresponds to the limit case where a cut is placed in every other domain.

To illustrate the gain in application time and the improved parallelism of the double sweep with cuts, Figure 4 presents timelines of the solves performed during the application of the Schwarz iteration operator and the double sweep preconditioner, with and without cuts. The timelines for the combined double sweep algorithm mentioned at the end of Section 3.1 are illustrated on Figure 5. The combined application avoids redundant solves, while not being faster in a fully parallel implementation. The main difference is in the flow of information during the sweeps.

3.3. Symmetric Gauss-Seidel Preconditioners

Instead of simplifying the Schwarz operator in order to derive explicit expressions for its inverse like was done for double sweep algorithms, standard preconditioning techniques could also be directly applied to the full Schwarz operator. For example, a symmetric Gauss-Seidel (SGS) preconditioner $M_{SGS}^{-1} = \bar{L}_N^{-1} \bar{U}_N^{-1}$ (\bar{U}_N and \bar{L}_N are the upper and lower triangular parts, with a unit diagonal) can be considered for the full matrix $\mathcal{F}_N(N)$, as the factors can also be computed a priori with a recursion formula, e.g. for 5 subdomains:

$$\bar{L}_N^{-1} r = \begin{bmatrix} \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\mathcal{E}_1^b & \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{I} & 0 & 0 & 0 & 0 & 0 \\ \mathcal{B}_2^f \mathcal{E}_1^b & -\mathcal{B}_2^f & -\mathcal{E}_2^b & \mathcal{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{I} & 0 & 0 & 0 \\ -\mathcal{B}_2^f \mathcal{B}_3^f \mathcal{E}_1^b & \mathcal{B}_2^f \mathcal{B}_3^f & \mathcal{B}_3^f \mathcal{E}_2^b & -\mathcal{B}_3^f & -\mathcal{E}_3^b & \mathcal{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} & 0 \\ \mathcal{B}_2^f \mathcal{B}_3^f \mathcal{B}_4^f \mathcal{E}_1^b & -\mathcal{B}_2^f \mathcal{B}_3^f \mathcal{B}_4^f & -\mathcal{B}_3^f \mathcal{B}_4^f \mathcal{E}_2^b & \mathcal{B}_3^f \mathcal{B}_4^f & \mathcal{B}_4^f \mathcal{E}_3^b & -\mathcal{B}_4^f & -\mathcal{E}_4^b & \mathcal{I} \end{bmatrix} \begin{bmatrix} r_{1,2} \\ r_{2,1} \\ r_{2,3} \\ r_{3,2} \\ r_{3,4} \\ r_{4,3} \\ r_{4,5} \\ r_{5,4} \end{bmatrix};$$

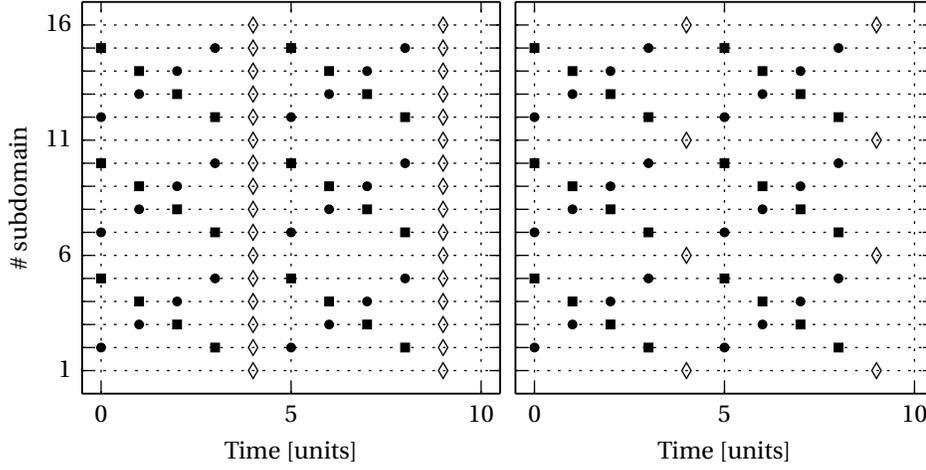


FIGURE 5. Illustration on timelines of the separate (left) and combined (right) application of the double sweep with cuts and the Schwarz operator, right-preconditioning. Black squares involve data on the right only (backward solves); black dots involve data on the left only (forward solves); white diamonds involve data on both sides (standard operator or cut domains).

$$\bar{U}_N^{-1}r = \begin{bmatrix} \mathcal{I} & -\mathcal{E}_2^f & -\mathcal{B}_2^b & \mathcal{B}_2^b \mathcal{E}_3^f & \mathcal{B}_2^b \mathcal{B}_3^b & -\mathcal{B}_2^b \mathcal{B}_3^b \mathcal{E}_4^f & -\mathcal{B}_2^b \mathcal{B}_3^b \mathcal{B}_4^b & \mathcal{B}_2^b \mathcal{B}_3^b \mathcal{B}_4^b \mathcal{E}_5^f \\ 0 & \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{I} & -\mathcal{E}_3^f & -\mathcal{B}_3^b & \mathcal{B}_3^b \mathcal{E}_4^f & \mathcal{B}_3^b \mathcal{B}_4^b & -\mathcal{B}_3^b \mathcal{B}_4^b \mathcal{E}_5^f \\ 0 & 0 & 0 & \mathcal{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{I} & -\mathcal{E}_4^f & -\mathcal{B}_4^b & \mathcal{B}_4^b \mathcal{E}_5^f \\ 0 & 0 & 0 & 0 & 0 & \mathcal{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} & -\mathcal{E}_5^f \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} r_{1,2} \\ r_{2,1} \\ r_{2,3} \\ r_{3,2} \\ r_{3,4} \\ r_{4,3} \\ r_{4,5} \\ r_{5,4} \end{bmatrix}.$$

The application of these factors to a vector is quite similar to the lower or upper parts of the standard double sweep matrix (11), although they have roughly twice as many non-zero elements for the forward or backward sweep. They give the recurrence relations for the applications $g' = \bar{L}_N^{-1}r$ or $g' = \bar{U}_N^{-1}r$:

$$\begin{aligned} g'_{2,1} &= r_{2,1} - \mathcal{E}_1^b r_{1,2}; \\ g'_{i+1,i} &= r_{i+1,i} - \mathcal{E}_i^b r_{i,i+1} - \mathcal{B}_i^f g'_{i,i-1}, \quad i = 2, \dots, N-1; \\ g'_{N-1,N} &= r_{N-1,N} - \mathcal{E}_N^f r_{N,N-1}; \\ g'_{i-1,i} &= r_{i-1,i} - \mathcal{E}_i^f r_{i,i-1} - \mathcal{B}_i^b g'_{i,i+1}, \quad i = N-1, \dots, 2. \end{aligned} \tag{18}$$

Comparing these relations with (13), we observe that a new term has appeared, that corresponds to a source on the same interface as the target interface. Consequently, the first solve can no longer be skipped, but the total number of solves per sweep is only increased by 1; the presence of the additional term essentially indicates that more information is incorporated in our sweeps, as we now account for reflections.

The recurrence relations can be written in the same form as (14), if the sweep definitions are slightly modified with the induction property:

$$\begin{aligned} \bar{S}_k^f(l) &= -\mathcal{E}_{I_k(l+1)}^b r_{I_k(l+1), I_k(l+2)} - \mathcal{B}_{I_k(l+1)}^f (r_{I_k(l+1), I_k(l)} + \bar{S}_k^f(l-1)); \\ \bar{S}_k^b(l) &= -\mathcal{E}_{I'_k(l+1)}^f r_{I'_k(l+1), I'_k(l+2)} - \mathcal{B}_{I'_k(l+1)}^b (r_{I'_k(l+1), I'_k(l)} + \bar{S}_k^b(l-1)), \end{aligned}$$

A significant difference with the standard double sweep is that the 2 sweeps are now performed sequentially instead of being parallel, hence with an application time that is doubled. Although the additive relation $\mathcal{F}_A^{-1} = (\bar{L}_A^{-1} + \bar{U}_A^{-1}) - I$ does not hold for \mathcal{F}_N , one can still design a preconditioner with the same additive structure: $M_{SGSa}^{-1} = (\bar{L}_N^{-1} + \bar{U}_N^{-1}) - I$. It was observed experimentally that this modification does not result in a significant degradation of the convergence rate, which makes it a better choice since its application time is reduced by a factor of 2. Strictly speaking, a copy of the input vector should be kept since the sweeps otherwise modify data that will be used by the other sweep after they have crossed. In practical implementations, it is however more convenient to ignore this and modify the data in-place in a more ‘‘Gauss-Seidel-like’’ fashion, which also resulted in faster convergence rates in our experiments. Compared to the standard double sweep preconditioner, this variant has demonstrated a convergence rate that is roughly doubled in problems where significant reflections occur, despite its very limited additional cost; this is especially true when less efficient transmission conditions are used, as we will see in the examples of Section 5.

3.4. Other Variants

Any other standard preconditioning technique could also be investigated along the lines of SGS and SGSa. For example, an SOR-like preconditioner with a relaxation factor ω could be constructed as follows:

$$M_{SOR}^{-1}(\omega) = (2 - \omega)((D + \omega L) + (D + \omega U)) - \omega I. \quad (19)$$

Our numerical experiments however concluded that the optimal relaxation parameter is actually $\omega = 1$, which reduces to SGSa. Incomplete LU factorizations could also be considered.

Introducing overlap in the (partial) sweeps would in turn lead to Additive Schwarz or Restricted Additive Schwarz preconditioners... of the non-overlapping Schwarz domain decomposition method.

4. ADDITIONAL PARALLELIZATION STRATEGIES

The sequential nature of the sweeping process can be alleviated when problems with multiple right-hand sides are considered, as is common in many geophysical exploration simulations, or in the electromagnetic simulation of antenna arrays. We pointed out earlier that applying the combination of the Schwarz operator and the sweeping preconditioner as a single routine has the effect of suppressing the ‘‘barrier’’ step where all subproblems are solved simultaneously, as illustrated on Figure 5. This gives room for the simultaneous execution of several sweeps acting on different right-hand sides in a pipelining fashion, see Figure 6, as a complementary way of avoiding CPU idleness when coupled with the cuts of the previous Section. The number of right-hand sides than can be efficiently pipelined depends on the length of the sweep, i.e. on the number of cuts.

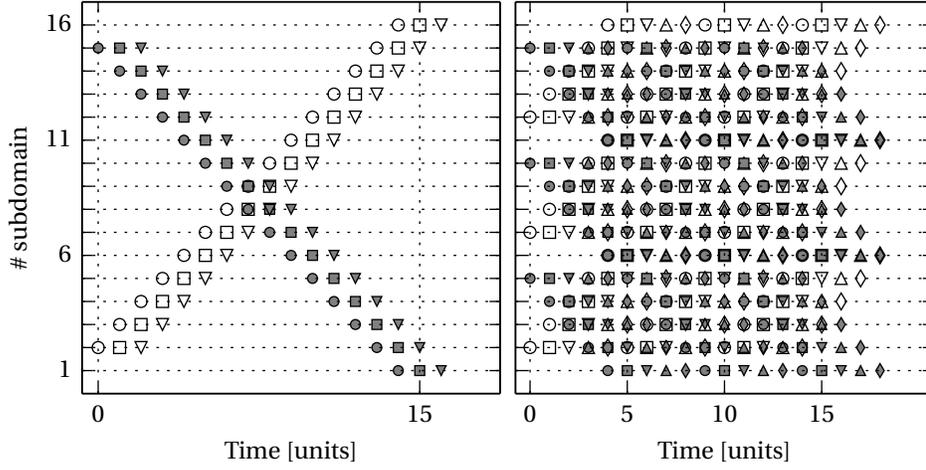


FIGURE 6. By pipelining multiple right-hand sides in the sweeps, CPU usage can be improved, while not increasing the duration of an iteration, except the last one. Left: 1 iteration with 3 right-hand sides, without cut; right: usage is maximized by pipelining as many RHS as the sweep length, here with 2 cuts. It is no coincidence that the same symbols (each corresponding to a given RHS) appear simultaneously on the cut domains, where the partial sweeps converge and for which one can solve a single problem.

We now study more in detail the algorithmic aspects of the combined double sweep with cuts. The application of operator \mathcal{F}_c on some residual vector r is illustrated below:

$$\begin{aligned}
 g' = \mathcal{F}_c r = & \begin{bmatrix} \mathcal{I} & \mathcal{E}_2^f & 0 & 0 & 0 & 0 \\ \mathcal{E}_1^b & \mathcal{I} & -\mathcal{E}_1^b \mathcal{B}_2^b & 0 & \mathcal{E}_1^b \mathcal{B}_2^b \mathcal{B}_3^b & 0 \\ 0 & -\mathcal{E}_3^f \mathcal{B}_2^f & \mathcal{I} & \mathcal{E}_3^f & 0 & 0 \\ 0 & 0 & \mathcal{E}_2^b & \mathcal{I} & -\mathcal{E}_2^b \mathcal{B}_3^b & 0 \\ 0 & \mathcal{E}_4^f \mathcal{B}_3^f \mathcal{B}_2^f & 0 & -\mathcal{E}_4^f \mathcal{B}_3^f & \mathcal{I} & \mathcal{E}_4^f \\ 0 & 0 & 0 & 0 & \mathcal{E}_3^b & \mathcal{I} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathcal{B}_4^b & 0 & -\mathcal{B}_4^b \mathcal{B}_5^b & 0 & \mathcal{B}_4^b \mathcal{B}_5^b \mathcal{B}_6^b & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_{1,2} \\ r_{2,1} \\ r_{2,3} \\ r_{3,2} \\ r_{3,4} \\ r_{4,3} \end{bmatrix} \\
 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{B}_4^f \mathcal{B}_3^f \mathcal{B}_2^f & 0 & -\mathcal{B}_4^f \mathcal{B}_3^f & 0 & \mathcal{B}_4^f \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathcal{I} & \mathcal{E}_5^f & 0 & 0 & 0 & 0 \\ \mathcal{E}_4^b & \mathcal{I} & -\mathcal{E}_4^b \mathcal{B}_5^b & 0 & \mathcal{E}_4^b \mathcal{B}_5^b \mathcal{B}_6^b & 0 \\ 0 & -\mathcal{E}_6^f \mathcal{B}_5^f & \mathcal{I} & \mathcal{E}_6^f & 0 & 0 \\ 0 & 0 & \mathcal{E}_5^b & \mathcal{I} & -\mathcal{E}_5^b \mathcal{B}_6^b & 0 \\ 0 & \mathcal{E}_7^f \mathcal{B}_6^f \mathcal{B}_5^f & 0 & -\mathcal{E}_7^f \mathcal{B}_6^f & \mathcal{I} & \mathcal{E}_7^f \\ 0 & 0 & 0 & 0 & \mathcal{E}_6^b & \mathcal{I} \end{bmatrix} \begin{bmatrix} r_{4,5} \\ r_{5,4} \\ r_{5,6} \\ r_{6,5} \\ r_{6,7} \\ r_{7,6} \end{bmatrix} \\
 = & \begin{bmatrix} r_{1,2} + \mathcal{E}_2^f r_{2,1} \\ r_{2,1} + \mathcal{E}_1^b r_{1,2} - \mathcal{E}_1^b \mathcal{B}_2^b r_{2,3} + \mathcal{E}_1^b \mathcal{B}_2^b \mathcal{B}_3^b r_{3,4} \\ r_{2,3} + \mathcal{E}_3^f r_{3,2} - \mathcal{E}_3^f \mathcal{B}_2^f r_{2,1} \\ r_{3,2} + \mathcal{E}_2^b r_{2,3} - \mathcal{E}_2^b \mathcal{B}_3^b r_{3,4} \\ r_{3,4} + \mathcal{E}_4^f r_{4,3} - \mathcal{E}_4^f \mathcal{B}_3^f r_{3,2} + \mathcal{E}_4^f \mathcal{B}_3^f \mathcal{B}_2^f r_{2,1} + \mathcal{B}_4^b r_{4,5} - \mathcal{B}_4^b \mathcal{B}_5^b r_{5,6} + \mathcal{B}_4^b \mathcal{B}_5^b \mathcal{B}_6^b r_{6,7} \\ r_{4,3} + \mathcal{E}_3^b r_{3,4} \\ r_{4,5} + \mathcal{E}_5^f r_{5,4} \\ r_{5,4} + \mathcal{E}_4^b r_{4,5} - \mathcal{E}_4^b \mathcal{B}_5^b r_{5,6} + \mathcal{E}_4^b \mathcal{B}_5^b \mathcal{B}_6^b r_{6,7} + \mathcal{B}_4^f r_{4,3} - \mathcal{B}_4^f \mathcal{B}_3^f r_{3,2} + \mathcal{B}_4^f \mathcal{B}_3^f \mathcal{B}_2^f r_{2,1} \\ r_{5,6} + \mathcal{E}_6^f r_{6,5} - \mathcal{E}_6^f \mathcal{B}_5^f r_{5,4} \\ r_{6,5} + \mathcal{E}_5^b r_{5,6} - \mathcal{E}_5^b \mathcal{B}_6^b r_{6,7} \\ r_{6,7} + \mathcal{E}_7^f r_{7,6} - \mathcal{E}_7^f \mathcal{B}_6^f r_{6,5} + \mathcal{E}_7^f \mathcal{B}_6^f \mathcal{B}_5^f r_{5,4} \\ r_{7,6} + \mathcal{E}_6^b r_{6,7} \end{bmatrix}.
 \end{aligned}$$

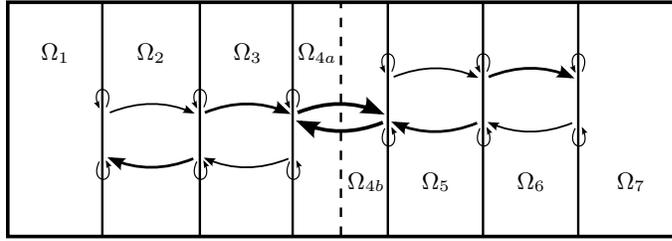


FIGURE 7. Illustration of the flow of information in the combined sweeping algorithm, with a cut.

The diagonal blocks of the matrix are combined sweeps over the groups of domains; off-diagonal blocks are the “bridges” between the groups (see Figure 7). We recall that the operators $\mathcal{B}_i^{\{f,b\}}$ and $\mathcal{E}_i^{\{f,b\}}$ actually correspond to the same solve of subdomain Ω_i : the difference is simply in their target interface. Since the cut domain needs to be solved with the same data in the forward and backward sweeps, a single solve must be performed with non-zero data on both sides, as opposed to the interior domains of each group.

The generic update expressions read:

$$g'_{i,i-1} = r_{i,i-1} + \mathcal{E}_{i-1}^b \left(r_{i-1,i} + S_k^b (I_k'(1) - i) \right) - \delta_{i,I_k(2)} S_{k-1}^f (N_{k-1} - 1), \quad i = 2, \dots, N;$$

$$g'_{i,i+1} = r_{i,i+1} + \mathcal{E}_{i+1}^f \left(r_{i+1,i} + S_k^f (i - I_k(1)) \right) - \delta_{i,I_k'(2)} S_{k+1}^b (N_{k+1} - 1), \quad i = 1, \dots, N - 1.$$

The Kronecker symbol δ determines whether the current domain is the last of the current sweep, in which case the contribution from the next group should be added. In practice, the algorithm works in a similar way than the preconditioner alone, by sweeping over the subdomains, while the flow of information is more complex: instead of ignoring the reflections, this data is now “left behind”, as is materialized by the presence of the $\mathcal{E}^{\{f,b\}}$ operators in the expressions. It is illustrated on Figure 7, to be compared with Figure 3.

Looking at the timing diagram for an iteration with a single RHS with the combined algorithm (Figure 5, right), one can observe long diagonals that are prolonged in the next iteration. This suggests that even in that case, pipelining multiple RHS is possible with little interference (it is obviously not in the separate case). When the sweeps cross each other, one should ideally be able to solve the same subproblem twice at the same time (a forward and a backward), regardless of the number of pipelined RHS, as illustrated on Figure 6.

5. NUMERICAL RESULTS

The results presented in this section were obtained with the GetDDM¹ framework [15], based on the open-source finite element mesh generator Gmsh [10] and finite element solver GetDP [5, 9].

5.1. Marmousi Model

The Marmousi model is a synthetic 2D acoustic model that reproduces the complex velocity profile of a slice of earth. It features a wide range of speeds, from 1500 m/s to 5500 m/s, with several layers and faults as depicted on Figure 8. It has become a classic test case for benchmarking seismic inversion codes [14]. We simulate the propagation of time-harmonic acoustic waves produced by a point source located right below the center of the top boundary (see Figure 9 for an illustration at a frequency of 40 Hz). Sommerfeld boundary conditions are imposed on all sides. A typical GMRES convergence history for the unpreconditioned and preconditioned domain decomposition algorithm is depicted on Figure 10 for a frequency of 400 Hz, with 352 subdomains. The finite element mesh is Cartesian, with about 750 million nodes, corresponding to 20 points per (smallest) wavelength in the domain. Figure 11 compares the convergence of the proposed algorithms in

¹<http://onelab.info/wiki/GetDDM>

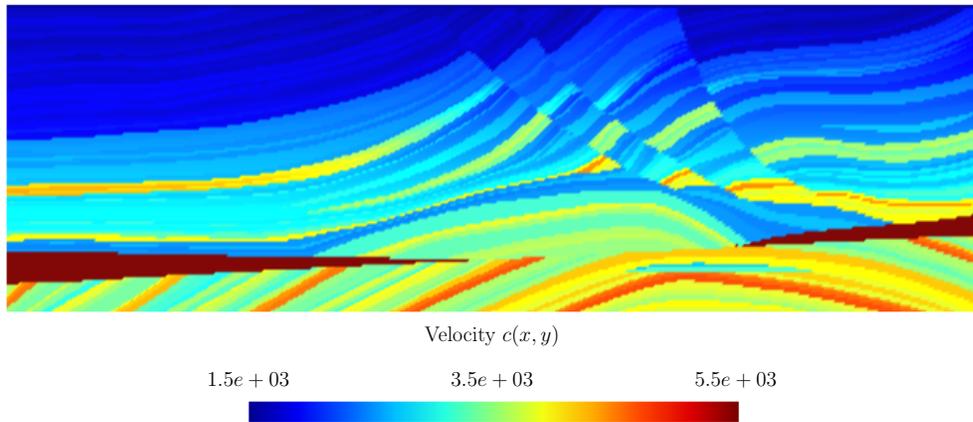


FIGURE 8. Velocity profile of the Marmousi model. Dimensions (in meters) are $[0, 9192] \times [0, -2904]$.

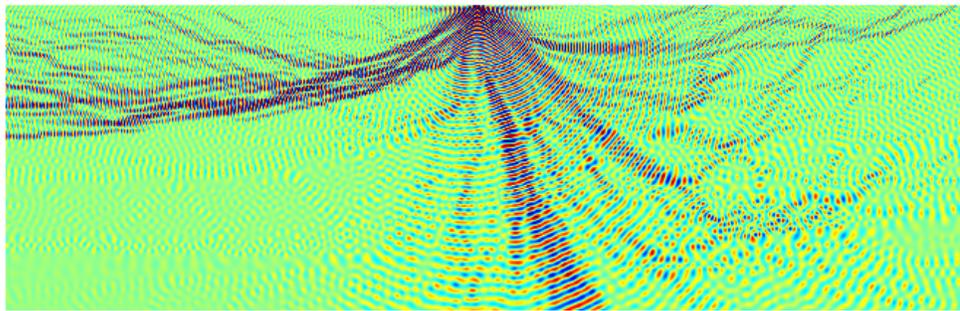


FIGURE 9. Solution of the Marmousi model at a frequency of 40 Hz, with a source located right below the center of the top boundary. Sommerfeld conditions are used on all boundaries.

the same situation, when varying the length of the sweeps. For a single RHS, short sweeps of about 15 to 20 subdomains are clearly the best option.

The analysis of Figure 10 is interesting as it gives an idea of the actual speed-up that can be expected by using the sweeping preconditioners and the pipelining. Solving for a single right-hand side without preconditioner takes approximately 1000 time units (defined as the time for a subdomain solve). Solving the same problem with the SGS preconditioner without cuts requires approximately $20 * 350 = 7000$ time units, which clearly shows the penalty due to CPU idleness during the sweeps. One should therefore use cuts to decrease the iterations duration: Figure 11 shows that the number of iterations remains constant for sweep lengths as short as 25 subdomains, leading to a duration of $20 * 25 = 500$. Hence a speed-up of 2 can be expected for a single right-hand side, while a significant amount of CPU idleness is still present.

Consider now the opposite situation where we want to solve for as many right-hand sides as there are subdomains; using full pipelining without cuts, it would take $N(2 * N_{it} + 4)^2$ time units to complete, while $N * N_{it} \approx 350.000$ time units would be necessary without preconditioner. The global speed-up is thus of the order of half the ratio of iteration counts between the two methods, which is approximately 25 in our example.

²We break down the time consumption of the pipelined algorithm as follows: the computation of the right-hand sides $\mathcal{F}_N^{-1}b$ for the combined algorithm takes an application of the sweeps. As the sweeps overlap, two subproblems must be solved at each step which doubles an iteration duration. Finally the roll-in and roll-out phases each take $N/2$ time units and once the problem is solved the solution must be computed in volume for each right-hand side. In total we have a duration of $2 * N * (N_{it} + 1) + N/2 + N/2 + N$ time units.

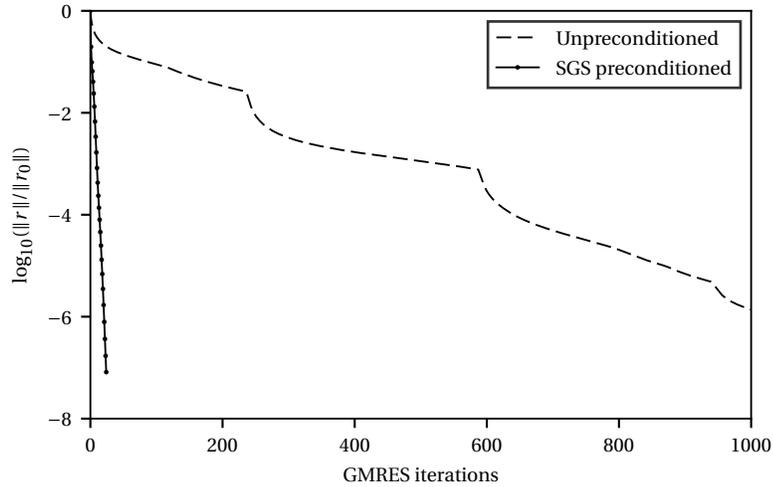


FIGURE 10. Typical convergence history for the Marmousi model at a frequency of 400 Hz, with or without preconditioning (352 subdomains, PML transmission condition).

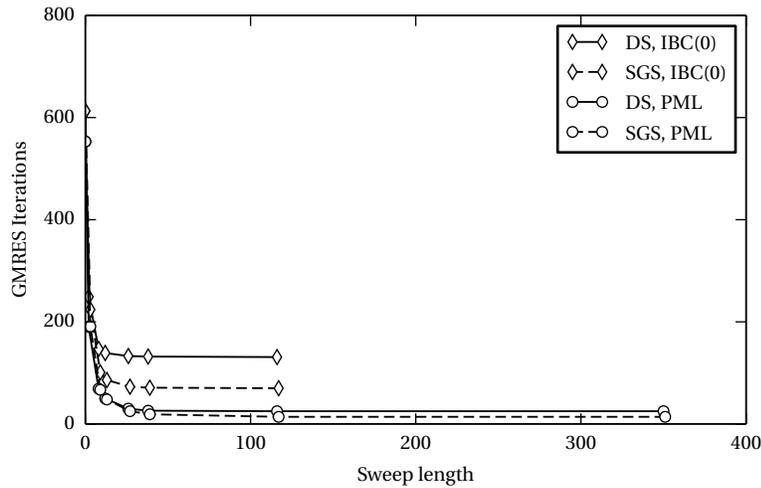


FIGURE 11. Number of GMRES iterations for the Marmousi model at a frequency of 400 Hz vs. sweep length. IBC(0) denotes simple Sommerfeld conditions on the artificial interfaces [15].

We note that the speed-up potential of pipelining for many right-hand sides can be boosted by increasing the number of subdomains, as the convergence rates of DS and SGS weakly depend on N , as opposed to the standard algorithm.

5.2. COBRA Model

We now present results obtained on a 3D S-shaped cavity: the COBRA benchmark defined by the JINA98 workgroup (see Figure 12). We solve this benchmark for both acoustic and electromagnetic waves at $k = 314\pi$ with a normalized wave speed of $c = 1$ m/s, with $N = 100$ subdomains and a finite element mesh with 20 points per wavelength. Figure 13 illustrate typical acoustic and electromagnetic solutions (at a lower frequency

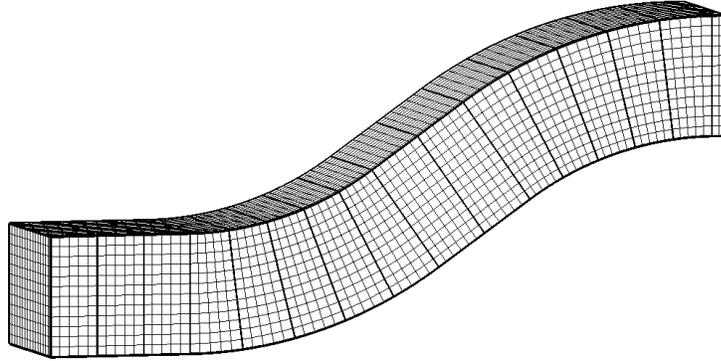


FIGURE 12. Mesh and decomposition of the cobra model. Dimensions are $0.084m \times 0.11m$ (cross-section); the curvilinear length from one end to the other is $0.469m$.

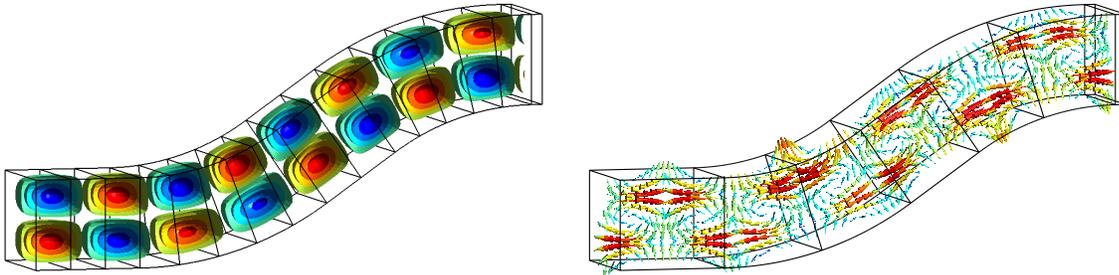


FIGURE 13. Acoustic solution (pressure isovalues, left) and electromagnetic solution (electric field on a transverse plane in the middle of the cross-section, right) of the COBRA model at $k = 30\pi$.

of $k = 30\pi$). Typical convergence results for the proposed algorithms are reported on Figure 14 for the acoustic case, and on Figure 15 for the electromagnetic case. The same conclusions as for the Marmousi model can be drawn: relatively short sweeps allow for a good compromise between iteration count and parallel efficiency.

6. CONCLUSION

We have presented several improvements to the double-sweep preconditioner for time-harmonic wave problems originally proposed in [18], which uses sweeping as a matrix-free preconditioner for a Schwarz domain decomposition method. The matrix representations of the Schwarz operator and its inverse were used to design these preconditioners: some of the new variants can be linked to well-known algebraic techniques like Block-Jacobi and Gauss-Seidel methods, which opens some perspectives as other techniques could be explored in a similar fashion. The introduction of cuts in the sweeps allows to restore some parallel efficiency to achieve some speed-up over the standard Schwarz method, which can be further improved by pipelining for problems with multiple right-hand sides. All these algorithms have the important property of being independent of the underlying problem formulation; we have presented results for the Helmholtz and Maxwell equations, in 2D and 3D, discretized using finite elements. The extension to elastodynamics is currently under investigation.

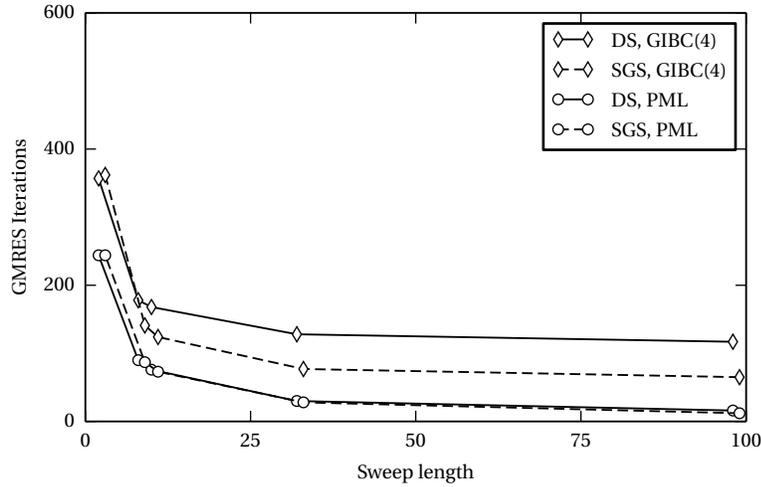


FIGURE 14. Number of GMRES iterations for the acoustic COBRA model at $k = 314\pi$ vs. sweep length. GIBC(4) denotes fourth order Padé impedance conditions on the artificial interfaces [15].

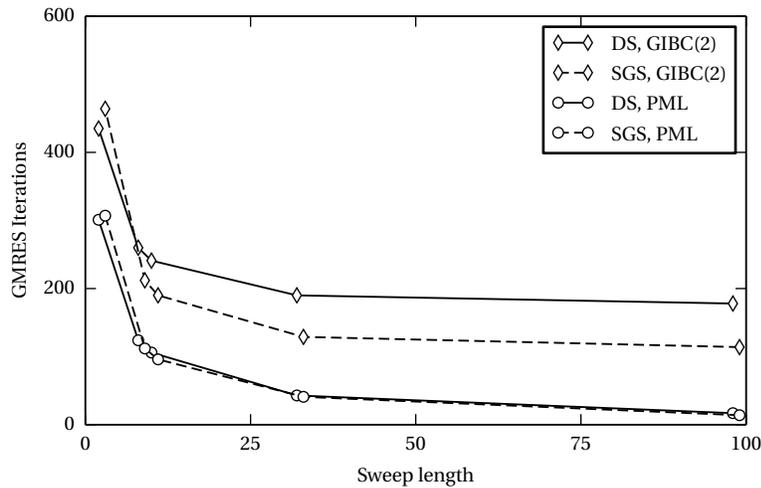


FIGURE 15. Number of GMRES iterations for the electromagnetic COBRA model at $k = 314\pi$ vs. sweep length. GIBC(2) denotes second order Padé impedance conditions on the artificial interfaces [15].

REFERENCES

- [1] J.-P. BERENGER, *A perfectly matched layer for the absorption of electromagnetic waves*, Journal of Computational Physics, 114 (1994), pp. 185–200.
- [2] A. BERMÚDEZ, L. HERVELLA-NIETO, A. PRIETO, AND R. RODRÍGUEZ, *An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems*, Journal of Computational Physics, 223 (2007), pp. 469–488.
- [3] M. E. BOUJAJI, B. THIERRY, X. ANTOINE, AND C. GEUZAIN, *A quasi-optimal domain decomposition algorithm for the time-harmonic Maxwell's equations*, Journal of Computational Physics, 294 (2015), pp. 38–57.

- [4] Y. BOUBENDIR, X. ANTOINE, AND C. GEUZAINÉ, *A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation*, Journal of Computational Physics, 231 (2012).
- [5] P. DULAR, C. GEUZAINÉ, F. HENROTTE, AND W. LEGROS, *A general environment for the treatment of discrete problems and its application to the finite element method*, IEEE Transactions on Magnetics, 34 (1998), pp. 3395–3398.
- [6] B. ENGQUIST AND L. YING, *Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation*, Communications on Pure and Applied Mathematics, 64 (2011), pp. 697–735.
- [7] ———, *Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers*, SIAM Journal on Multiscale Modeling & Simulation, 9 (2011), pp. 686–710.
- [8] M. GANDER, F. MAGOULÈS, AND F. NATAF, *Optimized Schwarz methods without overlap for the Helmholtz equation*, SIAM J. Sci. Comput., 24 (2002), pp. 38–60.
- [9] C. GEUZAINÉ, *GetDP: a general finite-element solver for the de Rham complex*, PAMM, 7 (2007), pp. 1010603–1010604.
- [10] C. GEUZAINÉ AND J.-F. REMACLE, *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering, 79 (2009), pp. 1309–1331.
- [11] A. MODAVE, E. DELHEZ, AND C. GEUZAINÉ, *Optimizing perfectly matched layers in discrete contexts*, International Journal for Numerical Methods in Engineering, 99 (2014), pp. 410–437.
- [12] J. POULSON, B. ENGQUIST, S. LI, AND L. YING, *A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations*, SIAM Journal on Scientific Computing, 35 (2013), pp. C194–C212.
- [13] V. RAWAT AND J.-F. LEE, *Nonoverlapping domain decomposition with second order transmission condition for the time-harmonic Maxwell's equations*, SIAM Journal on Scientific Computing, 32 (2010), pp. 3584–3603.
- [14] C. STOLK, *A rapidly converging domain decomposition method for the Helmholtz equation*, Journal of Computational Physics, 241 (2013), pp. 240 – 252.
- [15] B. THIERRY, A. VION, S. TOURNIER, M. E. BOUJAJI, D. COLIGNON, X. ANTOINE, AND C. GEUZAINÉ, *GetDDM: an open framework for testing optimized Schwarz methods for time-harmonic wave problems*, Computer Physics Communications, 203 (2016), pp. 309–330.
- [16] A. VION, *Multi-Domain Approaches for the solution of high-frequency time-harmonic propagation problems*, PhD thesis, University of Liège, Dec. 2014.
- [17] A. VION AND C. GEUZAINÉ, *Parallel double sweep preconditioner for the optimized Schwarz algorithm applied to high frequency Helmholtz and Maxwell equations*, in Proceedings of the 22th International Conference on Domain Decomposition Methods (DD22), Lugano, Italy, Sept. 2013.
- [18] ———, *Double sweep preconditioner for optimized Schwarz methods applied to the Helmholtz problem*, Journal of Computational Physics, 266 (2014), pp. 171–190.