Optimal Interface Conditions for an Arbitrary Decomposition into Subdomains

Martin J. Gander and Felix Kwok

Section de mathématiques, Université de Genève, Geneva CH-1211, Switzerland,
Martin.Gander@unige.ch; Felix.Kwok@unige.ch

Summary. The use of Dirichlet-to-Neumann operators as transmission conditions is known to yield optimal Schwarz methods that converge in a finite number of iterations when the subdomain decomposition has tree-like connectivity. However, it remains an open problem whether it is possible to construct a finitely terminating algorithm for arbitrary decompositions. In this article, we construct a Schwarz method that converges in exactly two steps for any decomposition into subdomains with minimal overlap. In this method, every subdomain must communicate with all other subdomains, but only data along subdomain boundaries need to be exchanged.

1 Optimal Interface Conditions

The convergence rate of Schwarz-type domain decomposition methods is very sensitive to the transmission condition used. Thus, it is natural to ask, for a given PDE and a given decomposition into subdomains, whether there exists a set of optimal interface conditions that leads to convergence in a finite number of steps. For a decomposition into vertical strips, we know that the Dirichlet-to-Neumann (DtN) operators yield such an optimal algorithm, see [4, 5, 6]. A similar result for decompositions whose connectivity graph contains no cycles is shown in [7]. It remains an open question to show whether similar operators exist for arbitrary decompositions.

The goal of this paper is to show that such an operator exists, at least in the discrete case, if we allow global communication between the subdomains, i.e., if each subdomain has access to the interface values of all the other subdomains. More precisely, we construct a subdomain iteration that converges to the exact solution in two steps by exchanging only data along subdomain boundaries.

We note that in general, these optimal interface conditions are nonlocal pseudo-differential operators, which are difficult to use in practice. Thus, the algorithm presented here is not meant to be implemented in a practical solver. However, practical algorithms can by derived by approximating the optimal operators by differential operators, see [3, 5] as well as [2] and references therein. Thus, our results serve as a starting point for this approximation process.

2 Notation and Assumptions

Let $\Omega \subset\subset \mathbb{R}^n$ be an open set. Suppose we want to solve the elliptic PDE

$$\mathcal{L}u = f \quad \text{on } \Omega, \qquad u = g \quad \text{on } \partial\Omega \tag{1}$$

by discretizing it to obtain the non-singular system $A\mathbf{u} = \mathbf{f}$ and using a domain decomposition method. Let Σ be the degrees of freedom therein. Suppose Ω is subdivided into nonoverlapping subdomains $\tilde{\Omega}_j$, $j = 1, \ldots, N$, and let $\tilde{\Sigma}_j$ be the discrete degrees of freedom contained within $\tilde{\Omega}_j$. Let $\{\Omega_j\}_{j=1}^N$ be an *overlapping* decomposition with degrees of freedom Σ_j, such that $\tilde{\Omega}_j \subset \Omega_j$ (and correspondingly $\tilde{\Sigma}_j \subset \Sigma_j$), and let R_j and $R_{\neg j}$ be operators that restrict Σ onto Σ_j and $\Sigma \setminus \Sigma_j$ respectively. We then define, for each $l = 1, \ldots, N$, the operator $\tilde{R}_l$, which has the same size as R_l, such that

$$[\tilde{R}_l]_{ij} = \begin{cases} 1 & \text{if } [R_l]_{ij} = 1 \text{ and } j \in \tilde{\Sigma}_l, \\ 0 & \text{otherwise.} \end{cases}$$

For each $j = 1, \ldots, N$, we define the matrices

$$A_j = R_j A R_j^T, \quad B_j = R_j A R_{\neg j}^T, \quad C_j = R_{\neg j} A R_j^T, \quad D_j = R_{\neg j} A R_{\neg j}^T.$$

We assume that D_j is nonsingular for all j, so that the Schur complement $A_j - B_j D_j^{-1} C_j$ is well-defined and non-singular. We now state the main assumption that will be used throughout the paper.

Assumption 1 (Sufficient Overlap) For all $j = 1, \ldots, N$, we have

$$\tilde{R}_j^T (R_j A - A_j R_j) = 0. \tag{2}$$

Assumption 1 states that the overlapping subdomain Σ_j needs to be sufficiently large, so that if v is a degree of freedom in $\tilde{\Sigma}_j$, then its stencil does not extend beyond Σ_j. This assumption is easily satisfied if the PDE is discretized using a compact stencil, because we can always construct Σ_j (and hence Ω_j) based on $\tilde{\Sigma}_j$ by extending it to include all points touched by the stencil.

3 Construction of the Method

The first step in constructing the method is to observe that the exact subdomain solution $\mathbf{u}_j = R_j \mathbf{u}$ can be obtained by solving the *Schur complement system*

$$(A_j - B_j D_j^{-1} C_j)\mathbf{u}_j = R_j \mathbf{f} - B_j D_j^{-1} R_{\neg j} \mathbf{f}. \tag{3}$$

If each subdomain has access to the right-hand side of all the other subdomains, then in principle we would be able to obtain $\mathbf{u}_j$ in one pass by solving each Schur complement system independently. However, this would not lead to an optimal Schwarz

method, because Schwarz methods only exchange information on $\mathbf{u}_i$ along subdomain boundaries. Thus, to construct an optimal Schwarz method, we must try to recover $R_{\neg j}\mathbf{f}$ using subdomain solutions only.

To do so, let us examine more closely what happens when we solve (3). First, we rewrite (3) using the definitions of A_j, B_j:

$$R_j A(R_j^T - R_{\neg j}^T D_j^{-1} C_j)\mathbf{u}_j = R_j\mathbf{f} - R_j A R_{\neg j}^T D_j^{-1} R_{\neg j}\mathbf{f}. \tag{4}$$

If we multiply (4) from the left by $\tilde{R}_j^T$, then the sufficient overlap assumption (2) implies that $\tilde{R}_j^T R_j A = \tilde{R}_j^T A_j R_j$. Thus, we get

$$\tilde{R}_j^T A_j R_j (R_j^T - R_{\neg j}^T D_j^{-1} C_j)\mathbf{u}_j = \tilde{R}_j^T R_j\mathbf{f} - \tilde{R}_j^T A_j R_j R_{\neg j}^T D_j^{-1} R_{\neg j}\mathbf{f}.$$

Since $R_j R_j^T = I$ and $R_j R_{\neg j}^T = 0$ (they restrict to Σ_j and $\Sigma \setminus \Sigma_j$ respectively, which are disjoint sets), the above equation simplifies to

$$\tilde{R}_j^T A_j \mathbf{u}_j = \tilde{R}_j^T R_j\mathbf{f}. \tag{5}$$

This means if $\mathbf{u}_j$ is the solution of (3), it is always possible to reconstruct $\tilde{R}_j\mathbf{f}$, the portion of $\mathbf{f}$ located in the *nonoverlapping* part of the subdomain, using only the subdomain solution $\mathbf{u}_j$. Since $\mathbf{f} = \sum_{i=1}^N \tilde{R}_i^T R_i\mathbf{f}$ and $R_{\neg j}\tilde{R}_j^T = 0$, we can substitute these relations into (3) to obtain the following algorithm.

Algorithm 1 For $k = 1, 2, \ldots$, and for $j = 1, \ldots, N$, solve

$$(A_j - B_j D_j^{-1} C_j)\mathbf{u}_j^{k+1} = R_j\mathbf{f} - \textstyle\sum_{i\neq j} T_{ji}\mathbf{u}_i^k, \tag{6}$$

where $T_{ji} = B_j D_j^{-1} R_{\neg j}\tilde{R}_i^T A_i$ is the transmission operator from Σ_i to Σ_j.

Theorem 1. *Let* $\mathbf{u}$ *be the exact solution to the problem* $A\mathbf{u} = \mathbf{f}$. *Then for any initial guess* $\mathbf{u}_j^0$, *Algorithm 1 converges to the exact solution in at most two iterations, i.e.,* $\mathbf{u}_j^2 = R_j\mathbf{u}$.

Proof. Since $\mathbf{u}_j^1$ is the solution of (3), by (5) we have $\tilde{R}_j^T A_j \mathbf{u}_j^1 = \tilde{R}_j^T R_j\mathbf{f}$. So the second step of Algorithm 1 gives

$$\begin{aligned}
(A_j - B_j D_j^{-1} C_j)\mathbf{u}_j^2 &= R_j\mathbf{f} - \sum_{i\neq j} B_j D_j^{-1} R_{\neg j}\tilde{R}_i^T A_i \mathbf{u}_i^1 \\
&= R_j\mathbf{f} - B_j D_j^{-1} R_{\neg j} \sum_{i\neq j} \tilde{R}_i^T R_i\mathbf{f} \\
&= R_j\mathbf{f} - B_j D_j^{-1} R_{\neg j} \sum_{i=1}^{N} \tilde{R}_i^T R_i\mathbf{f} \qquad (\text{since } R_{\neg j}\tilde{R}_j^T = 0) \\
&= R_j\mathbf{f} - B_j D_j^{-1} R_{\neg j}\mathbf{f},
\end{aligned}$$

which is exactly the Schur complement formulation of the system with the correct right hand side. This implies $\mathbf{u}_j^2 = R_j\mathbf{u}$, as required.

We now compare Algorithm 1 with the well-known parallel Schwarz method with optimal transmission conditions for the *tree case*:

$$(A_j - B_j D_j^{-1} C_j)\mathbf{u}_j^{k+1} = R_j\mathbf{f} - \sum_{(i,j)\in E}(B_j R_{\neg j} R_i^T + B_j D_j^{-1} C_j R_j R_i^T)\mathbf{u}_i^k,$$

where the sum is over all Ω_i that are neighbors of Ω_j. We know that the classical optimal algorithm only converges after $D+1$ iterations, where D is the diameter of the connectivity graph, see [4, 6, 7]. In contrast, Theorem 1 shows that Algorithm 1 will converge in at most two iterations, regardless of the number of subdomains and the topology of the decomposition. This comes at a cost: Algorithm 1 requires global communication among subdomains at every iteration, unlike its classical counterpart, which only requires communication between neighbors. Finally, we will show numerically in Sect. 5 that the classical algorithm can fail to converge when the decomposition is not a tree, while Algorithm 1 converges for decompositions with arbitrary connectivity.

4 Sparsity Pattern

Formula (6) seems to suggest at every step of Algorithm 1, every subdomain must have access to the entire solution in every other subdomain. This is in fact not the case. To understand which values really need to be transmitted, we study the sparsity pattern of T_{ji}, the operator through which subdomain j obtains information from $\mathbf{u}_i$. We show that this operator contains mostly zero columns, which means the corresponding nodal values are in fact discarded (and thus not needed). We first introduce the notion of the support of a vector.

Definition 1 (Support of a vector) Let $\mathbf{v}$ be a vector with degrees of freedom in Σ. Then the *support* of $\mathbf{v}$, denoted by $\mathrm{supp}(\mathbf{v})$, is the set of all points in Σ corresponding to nonzero entries in $\mathbf{v}$.

The following equivalences are immediate based on the definitions of supp:

(i) $\mathrm{supp}(\mathbf{v}) \subset \Sigma_j \iff R_j^T R_j \mathbf{v} = \mathbf{v} \iff R_{\neg j}\mathbf{v} = 0,$
(ii) $\mathrm{supp}(\mathbf{v}) \subset \tilde{\Sigma}_j \iff \tilde{R}_j^T R_j \mathbf{v} = \mathbf{v},$
(iii) $\mathrm{supp}(\mathbf{v}) \cap \Sigma_j = \emptyset \iff R_{\neg j}^T R_{\neg j}\mathbf{v} = \mathbf{v} \iff R_j\mathbf{v} = 0.$
(iv) $\mathrm{supp}(\mathbf{v}) \cap \tilde{\Sigma}_j = \emptyset \iff \tilde{R}_j\mathbf{v} = 0.$

Next, we identify the zero columns of T_{ji}. We do so by multiplying T_{ji} by a standard basis vector $\mathbf{e}_x$ and checking whether the product is zero.

Lemma 1. *Let $x \in \Sigma_i$ and $\mathbf{e}_x$ be its basis vector (1 at x and 0 everywhere else). Then we have $T_{ji}R_i\mathbf{e}_x = 0$ in each of the following cases:*

(i) $\mathrm{supp}(A\mathbf{e}_x) \cap \tilde{\Sigma}_i = \emptyset$;
(ii) $\mathrm{supp}(A\mathbf{e}_x) \subset \Sigma_j$;
(iii) $\{x\} \cup \mathrm{supp}(A\mathbf{e}_x) \subset \tilde{\Sigma}_i \setminus \Sigma_j$.

Proof. First, we rewrite $T_{ji}R_i\mathbf{e}_x$ as

$$T_{ji}R_i\mathbf{e}_x = B_jD_j^{-1}R_{\neg j}\tilde{R}_i^TA_iR_i\mathbf{e}_x = B_jD_j^{-1}R_{\neg j}\tilde{R}_i^TR_iA\mathbf{e}_x$$

by the sufficient overlap condition. We then consider the three cases:

(i) $\mathrm{supp}(A\mathbf{e}_x)\cap\tilde{\Sigma}_i=\emptyset$: we have

$$T_{ji}R_i\mathbf{e}_x = B_jD_j^{-1}R_{\neg j}(\tilde{R}_i^TR_iA\mathbf{e}_x) = B_jD_j^{-1}R_{\neg j}(R_i^T\underbrace{\tilde{R}_iA\mathbf{e}_x}_{=0}) = 0.$$

(ii) $\mathrm{supp}(A\mathbf{e}_x)\subset\Sigma_j$:

We have $\mathrm{supp}(\tilde{R}_iR_i^TA\mathbf{e}_x)\subset\mathrm{supp}(A\mathbf{e}_x)\subset\Sigma_j$, so $R_{\neg j}\tilde{R}_iR_i^TA\mathbf{e}_x=0$.

(iii) $\{x\}\cup\mathrm{supp}(A\mathbf{e}_x)\subset\tilde{\Sigma}_i\setminus\Sigma_j$:

Since $\mathrm{supp}(A\mathbf{e}_x)\subset\tilde{\Sigma}_i$, we have $\tilde{R}_iR_i^TA\mathbf{e}_x=A\mathbf{e}_x$. Now consider the linear system $D_j\mathbf{y}=R_{\neg j}A\mathbf{e}_x$, which can be rewritten as $R_{\neg j}AR_{\neg j}^T\mathbf{y}=R_{\neg j}A\mathbf{e}_x$. Since $x\notin\Sigma_j$, we see that $\mathbf{y}=R_{\neg j}\mathbf{e}_x$ satisfies the equation (because $R_{\neg j}^TR_{\neg j}\mathbf{e}_x=\mathbf{e}_x$). This is also the unique solution because D_j is nonsingular. Thus, we have $D_j^{-1}R_{\neg j}\tilde{R}_i^TR_iA\mathbf{e}_x=D_j^{-1}R_{\neg j}A\mathbf{e}_x=R_{\neg j}\mathbf{e}_x$. Now we multiply from the left by B_j to obtain

$$T_{ji}R_i\mathbf{e}_x = \underbrace{R_jAR_{\neg j}^T}_{B_j}R_{\neg j}\mathbf{e}_x = R_jA\mathbf{e}_x = 0,$$

since $\mathrm{supp}(A\mathbf{e}_x)$ lies completely outside Σ_j.

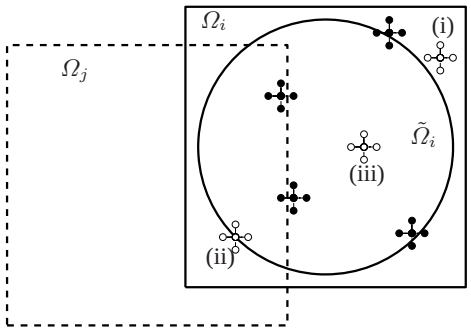

Fig. 1. A sketch showing stencils associated with different points in Ω_i. Stencils with solid nodes indicate points x at which $T_{ji}R_i\mathbf{e}_x\neq 0$; those with hollow nodes indicate points at which $T_{ji}R_i\mathbf{e}_x=0$ for the three cases indicated in Lemma 1.

Each of the three cases in Lemma 1 is illustrated in Fig. 1, where the hollow stencils indicate points that get mapped to zero by T_{ji}. Case (i) (top right-hand corner) happens when the stencil falls completely outside $\tilde{\Omega}_i$; case (ii) (bottom-left of

Ω_i) occurs when the stencil falls entirely within Ω_j. Finally, case (iii) occurs when the stencil is completely inside $\tilde{\Omega}_i \setminus \Omega_j$, just like the stencil near the center of the graphic. Thus, we see from Fig. 1 that the only points with $T_{ji}R_i\mathbf{e}_x \neq 0$ are those indicated by solid nodes, i.e. those that are so close to the boundary of $\tilde{\Omega}_i \setminus \Omega_j$ that their stencils straddle the boundary. These, in fact, are the only nodal values that must be transmitted. For a five-point stencil, this corresponds to a layer with a thickness of two nodes (one on each side of the boundary, see Fig. 2 in the next section); for wider stencils, e.g., for higher-order equations, this layer becomes thicker, but the number of values transmitted is still proportional to the length of $\partial(\tilde{\Omega}_i \setminus \Omega_j)$, which is one dimension lower than the set of all nodes in $\tilde{\Omega}_i$. If we define P_{ji} to be the restriction operator from Σ_i to the set of boundary nodes along $\partial(\tilde{\Omega}_i \setminus \Omega_j)$, then we can rewrite Algorithm 1 as follows:

Algorithm 2 For $k = 1, 2, \ldots$, and for $j = 1, \ldots, N$, solve

$$(A_j - B_j D_j^{-1} C_j)\mathbf{u}_j^{k+1} = R_j\mathbf{f} - \textstyle\sum_{i\neq j} \tilde{T}_{ji}\tilde{\mathbf{u}}_i^k,$$

where $\tilde{T}_{ji} = B_j D_j^{-1} R_{\neg j} \tilde{R}_i^T A_i P_{ji}^T$ and $\tilde{\mathbf{u}}_i^k = P_{ji}\mathbf{u}_i^k$.

Remark Algorithms 1 and 2 have identical iterates if the same initial guesses are used. Their only difference is that the latter does not transmit data corresponding to zero columns in T_{ji}, i.e., data that would be discarded anyway. This reduces the communication costs by a factor of H/h, where h is the fine mesh parameter and H is the size of the subdomain.

5 Numerical Examples

In this section, we present two examples in which we compare the convergence behavior of Algorithm 2 with that of the classical parallel Schwarz method with optimal transmission conditions, which is known to converge in a finite number of iterations in the tree case. For simplicity, in both cases we solve the 2D Poisson equation with Dirichlet boundary conditions, using the standard 5-point discretization. However, since the methods are derived purely algebraically, they are in principle applicable to any discretized PDE, provided we can define the subdomains so that they satisfy the sufficient overlap assumption, and that the subdomain problems are well posed.

Example 1 Here we decompose a rectangular domain into 6 vertical strips, as shown in Fig. 2a. Since the diameter of the connectivity graph is $D = 5$, we know that the parallel Schwarz method with optimal transmission conditions will converge in at most 6 steps; this is verified by the numerical results shown in Table 1. In contrast, Algorithm 2 converges in exactly two steps; this is in agreement with Theorem 1. Finally, Fig. 2a shows the communication pattern for both algorithms. As predicted by Lemma 1, the only nodal values that need to be transmitted are located on either side of the subdomain boundaries. Also, whereas the classical algorithm only takes

information from its neighbors, Algorithm 2 communicates with every subdomain, which makes it possible to converge in two iterations.

Example 2 In this example, we use the 4×4 decomposition shown in Fig. 2b. Since the connectivity graph is no longer a tree, we can no longer expect optimal parallel Schwarz to converge after a finite number of steps. Indeed, we see from Table 1 that the iteration diverges. This happens because of two reasons. First, since there are points belonging to more than two subdomains (i.e., cross points), optimal parallel Schwarz actually applies redundant updates at these points, leading to divergence, see [1]. In addition, unlike the tree case, $\partial\Omega_j$ is divided among several subdomains, so the boundary values obtained by Ω_j are no longer the trace of a harmonic function; instead, they are the trace of a function that fails to be harmonic at the partition points. Despite these difficulties, Algorithm 2 still converges in two iterations; the operators $\tilde{T}_{ji}$ are able to extract the right interface information and combine them the right way for the method to converge.

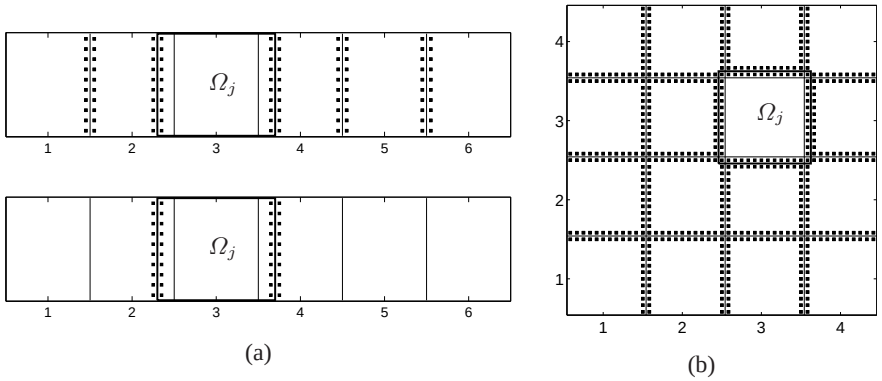

Fig. 2. Communication pattern for two decompositions into subdomains. *Black squares* indicate nodal values required by Ω_j, which is enclosed by *thick solid lines*. (**a**) decomposition into *vertical strips*. The *top figure* shows the values required by Algorithm 2, and the *bottom* those required by classical Parallel Schwarz with optimal transmission conditions. (**b**) a 4×4 decomposition, shown with the communication pattern for Algorithm 2.

6 Conclusion

We presented a new Schwarz method that converges in exactly two iterations when the domain decomposition satisfies the sufficient overlap assumption. Unlike the classical algorithm, the optimal transmission conditions we derived can handle arbitrary subdomain topologies. In our algorithm, each subdomain must communicate with all the other subdomains at each step; however, one only needs to exchange

Table 1. Parallel Schwarz with optimal transmission conditions versus Algorithm 2. In each case, we report the maximum L^∞ errors over all subdomains.

	Example 1 (6×1)		Example 2 (4×4)	
Its.	Parallel Schwarz	Algorithm 2	Parallel Schwarz	Algorithm 2
1	3.605×10^{0}	3.681×10^{0}	6.987×10^{1}	6.965×10^{1}
2	2.176×10^{-1}	1.066×10^{-14}	1.191×10^{2}	8.527×10^{-13}
3	1.252×10^{-2}		5.438×10^{1}	
4	7.328×10^{-4}		4.652×10^{2}	
5	3.278×10^{-5}		1.118×10^{3}	
6	1.066×10^{-14}		3.894×10^{3}	

data along a coarse grid structure containing the subdomain boundaries. Since its derivation is based only on sparse matrices, the method is in principle applicable to any PDE, or even systems of PDEs, as long as the subdomain problems remain solvable. Ongoing work focuses on deriving approximate local operators to obtain more efficient implementations.

References

1. E. Efstathiou and M.J. Gander. Why restricted additive Schwarz converges faster than additive Schwarz. *BIT*, 43(suppl.):945–959, 2003.
2. M.J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731 (electronic), 2006.
3. M.J. Gander, F. Magoules, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24:38–60, 2002.
4. F. Magoulès, F. Roux, and S. Salmon. Optimal discrete transmission conditions for a nonoverlapping domain decomposition method for the Helmholtz equation. *SIAM J. Sci. Comput.*, 25(5):1497–1515 (electronic), 2004.
5. F. Nataf and F. Rogier. Factorization of the convection-diffusion operator and the Schwarz algorithm. *Math. Models Methods Appl. Sci.*, 5:67–93, 1995.
6. F. Nataf, F. Rogier, and E. De Sturler. Optimal interface conditions for domain decomposition methods. Technical Report, École Polytech., Paris, 1994.
7. F. Nier. Remarques sur les algorithmes de décomposition de domaines. In *Seminaire: Équations aux Dérivées Partielles, 1998–1999*, Exp. No. IX, 26pp., Sémin. Équ. Dériv. Partielles. École Polytech., Palaiseau, 1999.