ADAPTIVE ALGEBRAIC OPTIMIZED SCHWARZ METHODS

CONOR MCCOID* AND FELIX KWOK†

Abstract. Optimized Schwarz methods use Fourier analysis to find transmission conditions between subdomains that provide faster convergence over standard Schwarz methods. However, this requires significant upfront analysis of the operator, and may not be straightforward for all problems. This work presents a new class of black box methods for adaptively optimizing the transmission conditions. This class of methods is shown to be part of the Krylov subspace family of methods. Analysis and examples show the effectiveness of these methods, especially in situations with multiple right hand sides for the same system.

Key words. domain decomposition, Krylov subspace, iterative methods, optimized Schwarz methods

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

1. Introduction.

Schwarz methods subdivide the domain of a problem into subdomains and solve the resulting subproblems repeatedly to iteratively solve the global problem. A core component of this iterative process is the transmission of information between the subdomains. This is done using transmission conditions, essentially boundary conditions that apply on the interface between subdomains.

The simplest transmission conditions are Dirichlet conditions, transmitting the solution at the interface directly from one subdomain to another. However, these come with limitations. There must be at least a minimal overlap between the subdomains or the iterative process will stagnate. Convergence is also suboptimal.

Neumann conditions allow for non-overlapping subdomains but can also be suboptimal. A combination of Neumann and Dirichlet conditions, Robin boundary conditions, has been shown to be significantly faster at converging all modes in the iterative process, assuming the Robin parameter is chosen well. Schwarz methods that use these transmission conditions are known as zero-th order optimized Schwarz methods (OSMs) [5]. Higher order OSMs can be constructed, in particular second order, but this can be cumbersome.

While OSMs are locally optimized, there exist optimal transmission conditions that reduce the iterative process to a finite number of steps, generally two. These are known as absorbing boundary conditions [6, 7, 9, 3]. Algebraically, they are equivalent to using the Schur complement. As such, they are prohibitively expensive to compute. Several workarounds have been proposed to approximate these transmission conditions in various applications [4, 26, 19, 1, 13].

In this paper, we develop a procedure to construct algebraic approximations to these absorbing boundary conditions through sequential rank one updates while also producing an accurate solution. The resulting conditions can be re-used in restarts, subsequent time steps, or similar problems to significantly reduce the number of iterations required. We show this procedure is a Krylov subspace method, and in particular it constructs a sequence of vectors numerically equivalent to those of GMRES.

A spectral version of this algorithm, using probing, has been studied in [12]. There, a set of probing vectors, usually representing Fourier modes expected to be

*Department of Mathematics and Statistics, McMaster University, Hamilton, ON L8S 4L8, Canada (mccoidc@mcmaster.ca).

†Department of Mathematics and Statistics, Université Laval, Québec, QC G1V 0H6, Canada (felix.kwok@mat.ulaval.ca).

problematic to the method, are decomposed into eigenvectors via the power iteration. The transmission conditions are then optimized with these vectors in mind.

2. Adaptive optimization.

2.1. Adaptive transmission conditions.

Consider a linear system of the form

$$\begin{bmatrix} A_{11} & A_{1\Gamma} & \\ A_{\Gamma 1} & A_{\Gamma\Gamma} & A_{\Gamma 2} \\ & A_{2\Gamma} & A_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_\Gamma \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \\ \mathbf{f}_2 \end{bmatrix}, \; A_{11} \in \mathbb{R}^{N_1 \times N_1}, \; A_{22} \in \mathbb{R}^{N_2 \times N_2}, \; A_{\Gamma\Gamma} \in \mathbb{R}^{M \times M}. \tag{2.1}$$

For large $N = N_1 + N_2 + M$, solving this system is costly in terms of both time and computational power. We break up the system into two subdomains which are solved iteratively using an algebraic Schwarz method. The two subproblems are

$$\begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T_{2\to 1}^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{n+1} \\ \mathbf{u}_{1\Gamma}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \end{bmatrix} - \begin{bmatrix} \\ A_{\Gamma 2}\mathbf{u}_2^n \end{bmatrix} + \begin{bmatrix} \\ T_{2\to 1}^{n+1}\mathbf{u}_{2\Gamma}^n \end{bmatrix}, \tag{2.2}$$

$$\begin{bmatrix} A_{22} & A_{2\Gamma} \\ A_{\Gamma 2} & A_{\Gamma\Gamma} + T_{1\to 2}^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_2^{n+1} \\ \mathbf{u}_{2\Gamma}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_\Gamma \end{bmatrix} - \begin{bmatrix} \\ A_{\Gamma 1}\mathbf{u}_1^n \end{bmatrix} + \begin{bmatrix} \\ T_{1\to 2}^{n+1}\mathbf{u}_{1\Gamma}^n \end{bmatrix}, \tag{2.3}$$

where $T_{1\to 2}^{n+1}$ and $T_{2\to 1}^{n+1}$ are transmission conditions between the subdomains. These conditions are allowed to change at each iteration, allowing them to adapt. We refer to the iterative solving of these systems with changing transmission conditions as adaptive optimized Schwarz methods (AOSMs).

Let us note here the difference between additive Schwarz methods and multiplicative Schwarz methods. In a multiplicative Schwarz method, the subproblems are solved sequentially, such that equation (2.2) is solved using $n = k$, then equation (2.3) is solved using $n = k + 1$, then equation (2.2) again using $n = k + 2$, and so on. In this way, n is always odd for equation (2.2) and always even for equation (2.3), or vice versa. In an additive Schwarz method, the subproblems are solved in parallel, such that both equations are solved for all values of n. While at first glance this appears faster, in most formulations this is equivalent to the sum of multiplicative non-interacting Schwarz methods and will therefore converge at the rate of the slowest of them. The advantage is its ability to be parallelized and produce a solution everywhere at any given iteration, albeit without improved precision.

If the transmission conditions are fixed, then the result is identically Schwarz methods. It is well established that under reasonable conditions the limit of $\mathbf{u}_i^n$ is $\mathbf{u}_i$ [5, 3]. As stated in the introduction, there are optimal choices of $T_{1\to 2}^{n+1}$ and $T_{2\to 1}^{n+1}$, namely

$$T_{1\to 2}^{n+1} = S_{1\to 2} := -A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma}, \quad T_{2\to 1}^{n+1} = S_{2\to 1} := -A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma},$$

the Schur complements which represent absorbing boundary conditions. There are also optimized transmission conditions used in OSMs, such as $T_{1\to 2}^{n+1} = T_{2\to 1}^{n+1} = -\frac{1}{2}A_{\Gamma\Gamma} + pI$, which are Robin boundary conditions for the Laplace equation discretized by finite differences on a uniform rectilinear grid [5, 9, 8].

One can reformulate the iterative process to construct a sequence of difference vectors that are added onto the initial guess. This is generally more stable numerically because the magnitude of the right hand side decreases as the solution approaches the limit. It will also be useful for our purposes to consider this sequence of vectors.

To reformulate equations (2.2) and (2.3), we consider the matrix for the $(n+1)$–th iterate applied to the difference between the $(n+1)$–th solution and the n–th solution:

$$\begin{aligned}
&\begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T_{2\to 1}^{n+1} \end{bmatrix} \left(\begin{bmatrix} \mathbf{u}_1^{n+1} \\ \mathbf{u}_{1\Gamma}^{n+1} \end{bmatrix} - \begin{bmatrix} \mathbf{u}_1^{n} \\ \mathbf{u}_{1\Gamma}^{n} \end{bmatrix} \right) \\
&= \begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T_{2\to 1}^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{n+1} \\ \mathbf{u}_{1\Gamma}^{n+1} \end{bmatrix} - \begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T_{2\to 1}^{n} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{n} \\ \mathbf{u}_{1\Gamma}^{n} \end{bmatrix} - \begin{bmatrix} \\ \Delta T_{2\to 1}^{n} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{n} \\ \mathbf{u}_{1\Gamma}^{n} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \end{bmatrix} - \begin{bmatrix} \\ A_{\Gamma 2}\mathbf{u}_2^{n} \end{bmatrix} + \begin{bmatrix} \\ T_{2\to 1}^{n+1}\mathbf{u}_{2\Gamma}^{n} \end{bmatrix} - \left(\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \end{bmatrix} - \begin{bmatrix} \\ A_{\Gamma 2}\mathbf{u}_2^{n-1} \end{bmatrix} + \begin{bmatrix} \\ T_{2\to 1}^{n}\mathbf{u}_{2\Gamma}^{n-1} \end{bmatrix} \right) \\
&\quad - \begin{bmatrix} \\ \Delta T_{2\to 1}^{n}\mathbf{u}_{1\Gamma}^{n} \end{bmatrix},
\end{aligned}$$

(2.4)
$$\begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T_{2\to 1}^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{d}_1^{n+1} \\ \mathbf{d}_{1\Gamma}^{n+1} \end{bmatrix} = - \begin{bmatrix} \\ A_{\Gamma 2}\mathbf{d}_2^{n} \end{bmatrix} + \begin{bmatrix} \\ T_{2\to 1}^{n+1}\mathbf{d}_{2\Gamma}^{n} \end{bmatrix} - \begin{bmatrix} \\ \Delta T_{2\to 1}^{n}(\mathbf{u}_{1\Gamma}^{n} - \mathbf{u}_{2\Gamma}^{n-1}) \end{bmatrix},$$

where $\Delta T_{i\to j}^n$ is the update to the transmission conditions $T_{i\to j}^n$, such that $T_{i\to j}^{n+1} = T_{i\to j}^n + \Delta T_{i\to j}^n$ and $\mathbf{d}_\nu^{n+1} = \mathbf{u}_\nu^{n+1} - \mathbf{u}_\nu^n$. Likewise, in the second subdomain,

(2.5)
$$\begin{bmatrix} A_{22} & A_{2\Gamma} \\ A_{\Gamma 2} & A_{\Gamma\Gamma} + T_{1\to 2}^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{d}_2^{n+1} \\ \mathbf{d}_{2\Gamma}^{n+1} \end{bmatrix} = - \begin{bmatrix} \\ A_{\Gamma 1}\mathbf{d}_1^{n} \end{bmatrix} + \begin{bmatrix} \\ T_{1\to 2}^{n+1}\mathbf{d}_{1\Gamma}^{n} \end{bmatrix} - \begin{bmatrix} \\ \Delta T_{1\to 2}^{n}(\mathbf{u}_{2\Gamma}^{n} - \mathbf{u}_{1\Gamma}^{n-1}) \end{bmatrix}.$$

Note that if the transmission conditions are fixed, then $\Delta T_{i\to j}^n$ is zero.

We will use the indices i and j to differentiate between the two subdomains. Thus, $i = 1$ or 2 and $j = 3 - i$. Equations that use i and j as indices apply in both subdomains.

Note that the top block rows in both equation (2.4) and (2.5) have null right hand sides. The corresponding vectors can therefore be eliminated:

$$\mathbf{d}_1^{n+1} = -A_{11}^{-1}A_{1\Gamma}\mathbf{d}_{1\Gamma}^{n+1}, \quad \mathbf{d}_2^{n+1} = -A_{22}^{-1}A_{2\Gamma}\mathbf{d}_{2\Gamma}^{n+1}. \tag{2.6}$$

This process is referred to in finite element method literature as static condensation or, more rarely, Guyan reduction [18]. The resulting equations, which define the sequence of differences exclusively on the interface between subdomains, are

$$\begin{aligned}
\left(A_{\Gamma\Gamma} + S_{1\to 2} + T_{2\to 1}^{n+1}\right)\mathbf{d}_{1\Gamma}^{n+1} &= \left(T_{2\to 1}^{n+1} - S_{2\to 1}\right)\mathbf{d}_{2\Gamma}^{n} - \Delta T_{2\to 1}^{n}\left(\mathbf{u}_{1\Gamma}^{n} - \mathbf{u}_{2\Gamma}^{n-1}\right), \\
\left(A_{\Gamma\Gamma} + S_{2\to 1} + T_{1\to 2}^{n+1}\right)\mathbf{d}_{2\Gamma}^{n+1} &= \left(T_{1\to 2}^{n+1} - S_{1\to 2}\right)\mathbf{d}_{1\Gamma}^{n} - \Delta T_{1\to 2}^{n}\left(\mathbf{u}_{2\Gamma}^{n} - \mathbf{u}_{1\Gamma}^{n-1}\right).
\end{aligned}$$

The sequence of differences is then dependent on the difference between the current transmission conditions, $T_{i\to j}^n$, and the optimal transmission conditions, $S_{i\to j}$. Let us denote this difference, the 'error' in the transmission conditions, as $E_{i\to j}^n = T_{i\to j}^n - S_{i\to j}$. Let us also denote $\hat{A} := A_{\Gamma\Gamma} + S_{1\to 2} + S_{2\to 1}$, which is the condensation of the matrix of equation (2.1) to the interface using Schur complements. The previous equations may then be written as

$$\left(\hat{A} + E_{2\to 1}^{n+1}\right)\mathbf{d}_{1\Gamma}^{n+1} = E_{2\to 1}^{n+1}\mathbf{d}_{2\Gamma}^{n} - \Delta T_{2\to 1}^{n}\left(\mathbf{u}_{1\Gamma}^{n} - \mathbf{u}_{2\Gamma}^{n-1}\right), \tag{2.7}$$

$$\left(\hat{A} + E_{1\to 2}^{n+1}\right)\mathbf{d}_{2\Gamma}^{n+1} = E_{1\to 2}^{n+1}\mathbf{d}_{1\Gamma}^{n} - \Delta T_{1\to 2}^{n}\left(\mathbf{u}_{2\Gamma}^{n} - \mathbf{u}_{1\Gamma}^{n-1}\right). \tag{2.8}$$

At this stage, $\Delta T^n_{i\to j}$ is unknown. We do not know yet how to adapt the transmission conditions in an optimal way.

We may also examine the evolution of the error in the solutions by applying the same process to the difference between the numerical and exact solutions:

$$\begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T^{n+1}_{2\to 1} \end{bmatrix} \left(\begin{bmatrix} \mathbf{u}_1^{n+1} \\ \mathbf{u}_{1\Gamma}^{n+1} \end{bmatrix} - \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_\Gamma \end{bmatrix} \right) = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \end{bmatrix} - \begin{bmatrix} \\ A_{\Gamma 2}\mathbf{u}_2^n \end{bmatrix} + \begin{bmatrix} \\ T^{n+1}_{2\to 1}\mathbf{u}^n_{2\Gamma} \end{bmatrix}$$

$$- \left(\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \end{bmatrix} - \begin{bmatrix} \\ A_{\Gamma 2}\mathbf{u}_2 \end{bmatrix} + \begin{bmatrix} \\ T^{n+1}_{2\to 1}\mathbf{u}_\Gamma \end{bmatrix} \right)$$

$$\begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T^{n+1}_{2\to 1} \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^{n+1} \\ \mathbf{e}_{1\Gamma}^{n+1} \end{bmatrix} = - \begin{bmatrix} \\ A_{\Gamma 2}\mathbf{e}_2^n \end{bmatrix} + \begin{bmatrix} \\ T^{n+1}_{2\to 1}\mathbf{e}^n_{2\Gamma} \end{bmatrix},$$

$$\begin{bmatrix} A_{22} & A_{2\Gamma} \\ A_{\Gamma 2} & A_{\Gamma\Gamma} + T^{n+1}_{1\to 2} \end{bmatrix} \begin{bmatrix} \mathbf{e}_2^{n+1} \\ \mathbf{e}_{2\Gamma}^{n+1} \end{bmatrix} = - \begin{bmatrix} \\ A_{\Gamma 1}\mathbf{e}_1^n \end{bmatrix} + \begin{bmatrix} \\ T^{n+1}_{1\to 2}\mathbf{e}^n_{1\Gamma} \end{bmatrix},$$

which may be condensed to

$$\left(\hat{A} + E^{n+1}_{2\to 1} \right) \mathbf{e}^{n+1}_{1\Gamma} = E^{n+1}_{2\to 1}\mathbf{e}^n_{2\Gamma}, \tag{2.9}$$

$$\left(\hat{A} + E^{n+1}_{1\to 2} \right) \mathbf{e}^{n+1}_{2\Gamma} = E^{n+1}_{1\to 2}\mathbf{e}^n_{1\Gamma}. \tag{2.10}$$

Assuming $\hat{A} + E^{n+1}_{i\to j}$ is invertible, there are two ways to reduce the magnitude of $\mathbf{e}^{n+1}_{j\Gamma}$: reducing the magnitude of $\mathbf{e}^n_{i\Gamma}$, and; reducing the product of $E^{n+1}_{i\to j}$ with $\mathbf{e}^n_{i\Gamma}$. In particular, if this product is zero, then the solution at the $(n+1)$–th iteration is exact. This is then the goal of adaptive transmission conditions, to increase the nullspace of $E^{n+1}_{i\to j}$.

2.2. Incremental approximations by low rank matrices. As stated above, if the matrix $E^n_{i\to j}$ is zero, then the Schwarz method will converge in the next iteration. We therefore seek ways to eliminate this matrix using information obtained during the runtime of the Schwarz method. While $E^n_{i\to j}$ is expensive to compute, the matrix-vector product $E^n_{i\to j}\mathbf{d}^n_{i\Gamma}$ is cheap. We can express this product as

$$E^n_{i\to j}\mathbf{d}^n_{i\Gamma} = \left(T^n_{i\to j} - S_{i\to j} \right) \mathbf{d}^n_{i\Gamma} = -A_{\Gamma i}\mathbf{d}^n_i + T^n_{i\to j}\mathbf{d}^n_{i\Gamma}, \tag{2.11}$$

where the last equality is due to equation (2.6).

Thus, we have a linear relation between two vectors that we do not wish to express explicitly. This represents an inverse problem: Given a vector and its image under a linear relation, find an approximation for the linear relation. The best that can be done is a rank one approximation. Suppose $\mathbf{y}_1 = E\mathbf{x}_1$ for some matrix E, then

$$E \approx \frac{\mathbf{y}_1\mathbf{x}_1^\top}{\|\mathbf{x}_1\|^2}. \tag{2.12}$$

Note the use of the Moore-Penrose pseudo-inverse of the vector $\mathbf{x}_1$ [20, 21].

Suppose instead that we have a linearly independent sequence of vectors $\{\mathbf{x}_k\}_{k=1}^n$ and the image of this sequence under the matrix $E \in \mathbb{R}^{M\times M}$, $\{\mathbf{y}_k\}_{k=1}^n$, such that

$$\mathbf{y}_k = E\mathbf{x}_k \quad \forall\ 1 \le k \le n. \tag{2.13}$$

Each vector pair gives its own rank one approximation of E, but the sum of these approximations does not form an approximation of E unless the vectors $\{\mathbf{x}_k\}$ are orthogonal. If we instead want an approximation of higher rank, we must orthogonalize

this set of vectors, and perform commenserate operations on the set $\{\mathbf{y}_k\}$. That is, if equation (2.12) is a rank one approximation of E, then let

$$\mathbf{w}_2 = \mathbf{x}_2 - \langle \mathbf{x}_1, \mathbf{x}_2 \rangle \mathbf{x}_1,$$
$$\mathbf{v}_2 = \mathbf{y}_2 - \langle \mathbf{x}_1, \mathbf{x}_2 \rangle \mathbf{y}_1.$$

This describes the modified Gram-Schmidt algorithm applied to the vectors $\{\mathbf{x}_k\}$, with an identical set of operations applied to $\{\mathbf{y}_k\}$. Then a rank two approximation of E is

$$E \approx \frac{\mathbf{y}_1 \mathbf{x}_1^\top}{\|\mathbf{x}_1\|^2} + \frac{\mathbf{v}_2 \mathbf{w}_2^\top}{\|\mathbf{w}_2\|^2}.$$

We formalize this process as Algorithm 2.1, which generates a low rank approximation of a matrix E. While the matrix E may be expensive to compute, we assume its matrix-vector products can be obtained relatively cheaply.

Algorithm 2.1 Iterative action approximation, $[V_n, W_n] = \mathrm{IAA}\left(\{\mathbf{x}_k\}_{k=1}^n, E\right)$

1: Inputs: $\{\mathbf{x}_k\}_{k=1}^n \subset \mathbb{R}^M$, $E \in \mathbb{R}^{M \times M}$
2: $\alpha_1 := 1/\|\mathbf{x}_1\|$, $\mathbf{w}_1 := \alpha_1 \mathbf{x}_1$, $\mathbf{v}_1 := \alpha_1 E \mathbf{x}_1$
3: **for** $k = 2 : n$ **do**
4: $\quad \mathbf{w}_k := \mathbf{x}_k$, $\mathbf{v}_k := E\mathbf{x}_k$
5: $\quad$ **for** $i = 1 : k-1$ **do**
6: $\quad\quad h \leftarrow \langle \mathbf{w}_i, \mathbf{w}_k \rangle$, $\mathbf{w}_k \leftarrow \mathbf{w}_k - h\mathbf{w}_i$
7: $\quad\quad \mathbf{v}_k \leftarrow \mathbf{v}_k - h\mathbf{v}_i$
8: $\quad \alpha_k := 1/\|\mathbf{w}_k\|$, $\mathbf{w}_k \leftarrow \alpha_k \mathbf{w}_k$, $\mathbf{v}_k \leftarrow \alpha_k \mathbf{v}_k$
9: Outputs: $W_n = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_n \end{bmatrix}$, $V_n = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \end{bmatrix}$
10: $V_n W_n^\top \approx E$

Line 7 implicitly updates the matrix E by removing the image of W_k from the output. At each iteration of the algorithm the rank one matrix $\mathbf{v}_k \mathbf{w}_k^\top$ is subtracted from E. This implicit update is useful to us and we seek to codify its properties in the following lemma.

LEMMA 2.1. *Let $\mathcal{S} = \{\mathbf{x}_k\}_{k=1}^n \subset \mathbb{R}^M$ be a set of linearly independent vectors, $E \in \mathbb{R}^{M \times M}$ a matrix. Let $[V_n, W_n] = \mathrm{IAA}(\mathcal{S}, E)$, as described in Algorithm 2.1, with columns $\mathbf{v}_k$ and $\mathbf{w}_k$, respectively. Let $E^1 := E$ and let $E^{k+1} := E^k - \mathbf{v}_k \mathbf{w}_k^\top$. Then*

$$E^{k+1} = E(I - W_k W_k^\top), \tag{2.14}$$
$$\mathbf{v}_k = E\mathbf{w}_k = E^k \mathbf{w}_k \tag{2.15}$$

for $1 \le k \le n$, where W_k is the first k columns of W_n.

Proof. We proceed by induction over k, starting with the base case at $k = 1$. We have that $\mathbf{y}_1 = E\mathbf{x}_1 = E^1\mathbf{x}$. Multiplying this equation by α_1 gives exactly equation (2.15) for $k = 1$. The matrix E^2 then satisfies

$$E^2 = E^1 - \mathbf{v}_1 \mathbf{w}_1^\top = E\left(I - \mathbf{w}_1 \mathbf{w}_1^\top\right),$$

thus proving equation (2.14) for $k = 1$.

Suppose the statements are true for $k < m$ and consider the case of $k = m$. By assumption, $\mathbf{y}_m = E\mathbf{x}_m$. The construction of $\mathbf{w}_m$, which is described in Algorithm 2.1 and is exactly the modified Gram-Schmidt algorithm, implies

$$\mathbf{x}_m = W_m \begin{bmatrix} \mathbf{h} \\ \alpha_m \end{bmatrix},$$

for some vector $\mathbf{h} \in \mathbb{R}^{m-1}$. Thus,

$$\begin{aligned} \mathbf{y}_m &= E^1 \mathbf{x}_m \\ &= E^1 W_m \begin{bmatrix} \mathbf{h} \\ \alpha_m \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_{m-1} & E^1 \mathbf{w}_m \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \alpha_m \end{bmatrix} \quad \text{by induction} \\ &= V_{m-1}\mathbf{h} + E^1 \mathbf{w}_m \alpha_m. \end{aligned}$$

Meanwhile, the construction of $\mathbf{v}_m$ in Algorithm 2.1 implies

$$\alpha_m \mathbf{v}_m = \mathbf{y}_m - V_{m-1}\mathbf{h}.$$

Combining these results gives the first equality of equation (2.15).

To prove equation (2.14), consider

$$\begin{aligned} E^{m+1} &= E^m - \mathbf{v}_m \mathbf{w}_m^\top \\ &= E\left(I - W_{m-1}W_{m-1}^\top\right) - \mathbf{v}_m \mathbf{w}_m^\top \quad \text{by induction} \\ &= E\left(I - W_{m-1}W_{m-1}^\top - \mathbf{w}_m \mathbf{w}_m^\top\right) \quad \text{by equation (2.15), first equality} \\ &= E\left(I - W_m W_m^\top\right) \end{aligned}$$

by orthogonality of $\{\mathbf{w}_k\}$. This then gives the second equality in equation (2.15). ∎

The matrix $I - W_n W_n^\top$ has some useful properties. It is symmetric, idempotent, singular with a nullspace equal to $\text{span}(W_n)$, and has $M-n$ non-zero eigenvalues equal to 1 with an eigenspace orthogonal to $\text{span}(W_n)$. Since this is true for all $n \leq M$, it is also true for the individual matrices $I - \mathbf{w}_n \mathbf{w}_n^\top$.

The implicit update to the linear relation E described in Theorem 2.1 will be our explicit update $\Delta T_{i\to j}^n$ to the transmission conditions:

$$\begin{aligned} \Delta T_{i\to j}^n &= T_{i\to j}^{n+1} - T_{i\to j}^n \\ &= E_{i\to j}^{n+1} - E_{i\to j}^n \\ &= -\mathbf{v}_i^n (\mathbf{w}_i^n)^\top, \end{aligned} \tag{2.16}$$

where the vectors $\mathbf{v}_i^n$ and $\mathbf{w}_i^n$ are the last columns of the matrices

$$[V_i^n, W_i^n] = \text{IAA}\left(\left\{\mathbf{d}_{i\Gamma}^k\right\}_{k=1}^n, E_{i\to j}^1\right), \quad \text{where } E_{i\to j}^1 \mathbf{d}_{i\Gamma}^k = -A_{\Gamma i}\mathbf{d}_i^k + T_{i\to j}^1 \mathbf{d}_{i\Gamma}^k. \tag{2.17}$$

By subtracting off $V_i^{k-1}(W_i^{k-1})^\top \mathbf{d}_{i\Gamma}^k$ from the product $E_{i\to j}^1 \mathbf{d}_{i\Gamma}^k$, as is done in the course of Algorithm 2.1, we retrieve equation (2.11). Thus, due to Theorem 2.1, we may use the two expressions interchangeably.

The update of equation (2.16) eliminates the contribution of $E_{i\to j}^{n+1}\mathbf{d}_{i\Gamma}^{n}$ from equations (2.7) and (2.8):

$$\begin{aligned}\left(\hat{A}+E_{i\to j}^{n+1}\right)\mathbf{d}_{j\Gamma}^{n+1} &= E_{i\to j}^{n+1}\mathbf{d}_{i\Gamma}^{n}-\Delta T_{i\to j}^{n}\left(\mathbf{u}_{j\Gamma}^{n}-\mathbf{u}_{i\Gamma}^{n-1}\right)\\ &= E_{i\to j}^{n}\mathbf{d}_{i\Gamma}^{n}+\Delta T_{i\to j}^{n}\mathbf{d}_{i\Gamma}^{n}-\Delta T_{i\to j}^{n}\left(\mathbf{u}_{j\Gamma}^{n}-\mathbf{u}_{i\Gamma}^{n-1}\right)\\ &= \left((\mathbf{w}_i^n)^\top\mathbf{d}_{i\Gamma}^{n}\right)\mathbf{v}_i^n-\mathbf{v}_i^n(\mathbf{w}_i^n)^\top\mathbf{d}_{i\Gamma}^{n}+\mathbf{v}_i^n(\mathbf{w}_i^n)^\top\left(\mathbf{u}_{j\Gamma}^{n}-\mathbf{u}_{i\Gamma}^{n-1}\right)\\ &= \mathbf{v}_i^n(\mathbf{w}_i^n)^\top\left(\mathbf{u}_{j\Gamma}^{n}-\mathbf{u}_{i\Gamma}^{n-1}\right),\end{aligned}$$

or, when presented as the full system, to

$$\begin{bmatrix}A_{jj} & A_{j\Gamma}\\ A_{\Gamma j} & A_{\Gamma\Gamma}+T_{i\to j}^{n+1}\end{bmatrix}\begin{bmatrix}\mathbf{d}_j^{n+1}\\ \mathbf{d}_{j\Gamma}^{n+1}\end{bmatrix}=(\mathbf{w}_i^n)^\top\left(\mathbf{u}_{j\Gamma}^{n}-\mathbf{u}_{i\Gamma}^{n-1}\right)\begin{bmatrix}\\ \mathbf{v}_i^n\end{bmatrix}. \tag{2.18}$$

2.3. AOSM algorithms. We may now write the AOSM algorithms using the update defined in equation (2.16). We define two types of AOSM: alternating AOSM (altAOSM), which modifies multiplicative Schwarz methods where subdomains are solved sequentially, and parallel AOSM (paraAOSM), which modifies additive Schwarz methods where subdomains are solved in parallel.

Algorithm 2.2 altAOSM: AOSM applied to multiplicative Schwarz

Start with initial transmission conditions $T_{1\to 2}^1$ and $T_{2\to 1}^1$
Make initial guess $\mathbf{u}_{1\Gamma}^0$
Calculate $\mathbf{u}_1^0 = A_{11}^{-1}(\mathbf{f}_1 - A_{1\Gamma}\mathbf{u}_{1\Gamma}^0)$
Solve equation (2.3) with $n = 0$
Solve equation (2.2) with $n = 1$
Calculate $\mathbf{d}_{1\Gamma}^2 = \mathbf{u}_{1\Gamma}^2 - \mathbf{u}_{1\Gamma}^0$ and $\mathbf{d}_1^2$ and set $n = 2$
while $\|\mathbf{d}_{1\Gamma}^n\| + \left\|\mathbf{d}_{2\Gamma}^{n-1}\right\| \geq tol$ **do**
    for $i = 1 : 2$ **do**
        Run an iteration of Algorithm 2.1, see equation (2.17)
        Set $\Delta T_{i\to j}^n = -\mathbf{v}_i^n(\mathbf{w}_i^n)^\top$, see equation (2.16)
        Solve equation (2.18)
        $\mathbf{u}_j^{n+1} := \mathbf{u}_j^{n-1} + \mathbf{d}_j^{n+1}$, $\mathbf{u}_{j\Gamma}^{n+1} := \mathbf{u}_{j\Gamma}^{n-1} + \mathbf{d}_{j\Gamma}^{n+1}$
        $n \leftarrow n + 1$
Output: $\mathbf{u} = [\mathbf{u}_1^n\ ;\ (\mathbf{u}_{1\Gamma}^n + \mathbf{u}_{2\Gamma}^{n-1})/2\ ;\ \mathbf{u}_2^{n-1}]$

Equation (2.18) can be replaced with equation (2.2) when $i = 2$ and equation (2.3) when $i = 1$. The former is a corrector formulation of the latter, as described in the preamble to equations (2.4) and (2.5). Using equation (2.18), one must add together the difference vectors to arrive at the solution vectors. Using equations (2.2) and (2.3), one must subtract sequential solution vectors to compute the difference vectors.

The differences between altAOSM and paraAOSM can be summarized by the schemas of Figure 2.1. In altAOSM, in the left of the figure, updates to the transmission conditions are generated through a single sequence of iterative subdomain solves. In paraAOSM, in the right of the figure, updates are generated by taking differences between two sequences of iterative solves.

3. Practical considerations.

Algorithm 2.3 paraAOSM: AOSM applied to additive Schwarz

Start with initial transmission conditions $T^1_{1\to 2}$ and $T^1_{2\to 1}$
Make initial guesses $\mathbf{u}^0_{1\Gamma}$ and $\mathbf{u}^0_{2\Gamma}$
Calculate $\mathbf{u}^0_1 = A_{11}^{-1}(\mathbf{f}_1 - A_{1\Gamma}\mathbf{u}^0_{1\Gamma})$ and equivalently $\mathbf{u}^0_2$
Solve equations (2.2) and (2.3) with $n = 0$
Calculate $\mathbf{d}^1_{1\Gamma}$, $\mathbf{d}^1_1$, $\mathbf{d}^1_{2\Gamma}$ and $\mathbf{d}^1_2$ and set $n = 1$
while $\|\mathbf{d}^n_{1\Gamma}\| + \|\mathbf{d}^n_{2\Gamma}\| \geq tol$ **do**
    for $i = 1 : 2$ **do**
        Run an iteration of Algorithm 2.1, see equation (2.17)
        Set $\Delta T^n_{i\to j} = -\mathbf{v}^n_i(\mathbf{w}^n_i)^\top$, see equation (2.16)
        Solve equation (2.18)
        $\mathbf{u}^{n+1}_j := \mathbf{u}^n_j + \mathbf{d}^{n+1}_j$, $\mathbf{u}^{n+1}_{j\Gamma} := \mathbf{u}^n_{j\Gamma} + \mathbf{d}^{n+1}_{j\Gamma}$
    $n \leftarrow n + 1$
Output: $\mathbf{u} = [\mathbf{u}^n_1 \ ; \ (\mathbf{u}^n_{1\Gamma} + \mathbf{u}^n_{2\Gamma})/2 \ ; \ \mathbf{u}^n_2]$

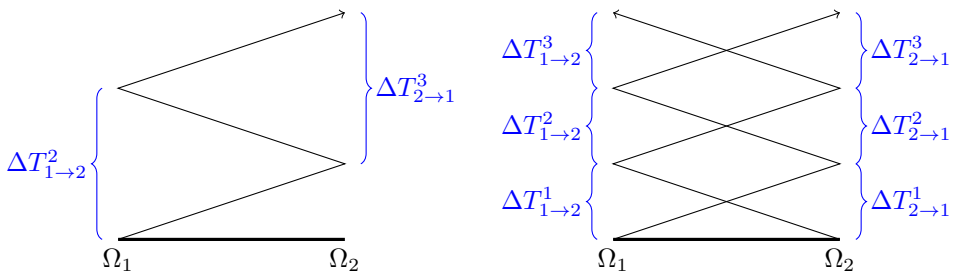

Fig. 2.1: Schema for calculating the updates to the transmission conditions in the AOSMs. (Left) altAOSM, Algorithm 2.2; (Right) paraAOSM, Algorithm 2.3.

3.1. Woodbury matrix identity. Updating the system matrices every iteration prevents the use of factorizations to speed up the system solves. Luckily, since the update to the matrices amounts to the addition of low rank matrices, we may update the solutions without updating the matrices. To accomplish this, we deploy the Woodbury matrix identity [27], a generalisation of the Sherman-Morrison formula [24].

PROPOSITION 3.1 (Woodbury matrix identity). *If* $\tilde{\mathbf{u}}$ *is the solution to* $A\tilde{\mathbf{u}} = \mathbf{b}$, $A \in \mathbb{R}^{N\times N}$, $\mathbf{b} \in \mathbb{R}^N$, *then*

$$\mathbf{u} = \tilde{\mathbf{u}} + A^{-1}V(I - W^\top A^{-1}V)^{-1}W^\top\tilde{\mathbf{u}} \tag{3.1}$$

is the solution to $(A - VW^\top)\mathbf{u} = \mathbf{b}$, *where* $V, W \in \mathbb{R}^{N\times k}$, $I \in \mathbb{R}^{k\times k}$.

Equation (2.18), which appears in both altAOSM and paraAOSM, may be represented as

((2.18) revisited)

$$\left(\begin{bmatrix} A_{jj} & A_{j\Gamma} \\ A_{\Gamma j} & A_{\Gamma\Gamma} + T^1_{i\to j}\end{bmatrix} - \begin{bmatrix} V^n_i \end{bmatrix}\begin{bmatrix} W^n_i \end{bmatrix}^\top\right)\begin{bmatrix} \mathbf{d}^{n+1}_j \\ \mathbf{d}^{n+1}_{j\Gamma}\end{bmatrix} = (\mathbf{w}^n_i)^\top\left(\mathbf{u}^n_{j\Gamma} - \mathbf{u}^{n-1}_{i\Gamma}\right)\begin{bmatrix} \mathbf{v}^n_i \end{bmatrix}.$$

Suppose we have some way to speed up the computation of vectors $\mathbf{z}^n_j$ and $\mathbf{z}^n_{j\Gamma}$ and

matrices Z_j^n and $Z_{j\Gamma}^n$ with columns $\mathbf{z}_j^n$ and $\mathbf{z}_{j\Gamma}^n$ in

$$\begin{bmatrix} A_{jj} & A_{j\Gamma} \\ A_{\Gamma j} & A_{\Gamma\Gamma} + T_{i\to j}^1 \end{bmatrix} \begin{bmatrix} \mathbf{z}_j^n \\ \mathbf{z}_{j\Gamma}^n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_i^n \end{bmatrix}, \quad \begin{bmatrix} A_{jj} & A_{j\Gamma} \\ A_{\Gamma j} & A_{\Gamma\Gamma} + T_{i\to j}^1 \end{bmatrix} \begin{bmatrix} Z_j^n \\ Z_{j\Gamma}^n \end{bmatrix} = \begin{bmatrix} V_i^n \end{bmatrix}, \tag{3.2}$$

such as a factorization, and we wish to use it to solve equation (2.18). Then, using Theorem 3.1,

$$\begin{aligned} \begin{bmatrix} \mathbf{d}_j^{n+1} \\ \mathbf{d}_{j\Gamma}^{n+1} \end{bmatrix} &= (\mathbf{w}_i^n)^\top \left(\mathbf{u}_{j\Gamma}^n - \mathbf{u}_{i\Gamma}^{n-1}\right) \left(\begin{bmatrix} \mathbf{z}_j^n \\ \mathbf{z}_{j\Gamma}^n \end{bmatrix} + \begin{bmatrix} Z_j^n \\ Z_{j\Gamma}^n \end{bmatrix} \left(I - (W_i^n)^\top Z_{j\Gamma}^n\right)^{-1} (W_i^n)^\top \mathbf{z}_{j\Gamma}^n \right) \\ &= (\mathbf{w}_i^n)^\top \left(\mathbf{u}_{j\Gamma}^n - \mathbf{u}_{i\Gamma}^{n-1}\right) \begin{bmatrix} Z_j^n \\ Z_{j\Gamma}^n \end{bmatrix} \left(I + \left(I - (W_i^n)^\top Z_{j\Gamma}^n\right)^{-1} (W_i^n)^\top Z_{j\Gamma}^n \right) \begin{bmatrix} \\ 1 \end{bmatrix} \\ &= (\mathbf{w}_i^n)^\top \left(\mathbf{u}_{j\Gamma}^n - \mathbf{u}_{i\Gamma}^{n-1}\right) \begin{bmatrix} Z_j^n \\ Z_{j\Gamma}^n \end{bmatrix} \left(I - (W_i^n)^\top Z_{j\Gamma}^n\right)^{-1} \begin{bmatrix} \\ 1 \end{bmatrix}. \end{aligned} \tag{3.3}$$

LEMMA 3.2. *Let $n \geq 1$. Let $\left\{\mathbf{d}_{j\Gamma}^k\right\}_{k=2}^{n+1}$ be the solutions of equation (2.18). Let $\left\{\mathbf{z}_{j\Gamma}^k\right\}_{k=1}^{n}$ be the solutions of equation (3.2), where $\left\{\mathbf{v}_i^k\right\}_{k=1}^{n}$ are the columns of V_i^n defined in equation (2.17). If these two sets of vectors are linearly independent, then*

$$\operatorname{span}\left(\left\{\mathbf{d}_{j\Gamma}^k\right\}_{k=2}^{n+1}\right) = \operatorname{span}\left(\left\{\mathbf{z}_{j\Gamma}^k\right\}_{k=1}^{n}\right).$$

Proof. We proceed by induction. The base case occurs for $n = 1$, when $V_i^n = \mathbf{v}_i^1$ and $W_i^n = \mathbf{w}_i^1$. Then equation (3.3) simplifies to

$$\begin{bmatrix} \mathbf{d}_j^2 \\ \mathbf{d}_{j\Gamma}^2 \end{bmatrix} = \beta \begin{bmatrix} \mathbf{z}_j^1 \\ \mathbf{z}_{j\Gamma}^1 \end{bmatrix},$$

where β is a non-zero scalar. Thus, $\mathbf{d}_{j\Gamma}^2$ is parallel to $\mathbf{z}_{j\Gamma}^1$.

Suppose the statement is true for $n-1$. By equation (3.3) $\mathbf{d}_{j\Gamma}^{n+1}$ lies in $\operatorname{span}(Z_{j\Gamma}^n)$. Let us denote

$$\mathbf{d}_{j\Gamma}^{n+1} = \sum_{k=1}^{n} b_k \mathbf{z}_{j\Gamma}^k,$$

for some constants b_k. By the assumption of linear independence, $b_n \neq 0$. By induction, there is a linear combination of the vectors $\mathbf{d}_{j\Gamma}^m$ for $2 \leq m \leq k+1$ such that

$$\mathbf{z}_{j\Gamma}^k = \sum_{m=2}^{k+1} a_{m,k} \mathbf{d}_{j\Gamma}^m.$$

We may now isolate for $\mathbf{z}_{j\Gamma}^n$ by subtracting all other vectors $\mathbf{z}_{j\Gamma}^k$ for $k < n$ from $\mathbf{d}_{j\Gamma}^{n+1}$:

$$\begin{aligned} b_n \mathbf{z}_{j\Gamma}^n &= \mathbf{d}_{j\Gamma}^{n+1} - \sum_{k=1}^{n-1} b_k \mathbf{z}_{j\Gamma}^k \\ &= \mathbf{d}_{j\Gamma}^{n+1} - \sum_{k=1}^{n-1} \sum_{m=2}^{k+1} b_k a_{m,k} \mathbf{d}_{j\Gamma}^m. \end{aligned}$$

Therefore, $\mathbf{z}_{j\Gamma}^n$ lies in the span of the vectors $\mathbf{d}_{j\Gamma}^k$. ∎

As a corollary to Theorem 3.2, one can replace $\mathbf{d}_{j\Gamma}^n$ in equation (2.17) with $\mathbf{z}_{j\Gamma}^{n-1}$ and still arrive at the same matrix W_j^{n+1}. This removes the update to the transmission conditions in the matrices on the left hand side of equations (2.2) and (2.3). Instead, only the transmission conditions on the right hand side are updated. This process is comparable to the Arnoldi method, in that the search directions are orthogonalized and the matrix remains constant.

3.2. Schur shuffle. The Woodbury matrix identity involves the inversion of $B_i^n = I_{n\times n} - (W_i^n)^\top Z_{j\Gamma}^n$. This is a matrix of size at most $n \times n$ and thus relatively small and inexpensive to invert. We can still simplify its calculation to make it as efficient as possible.

As n increments throughout an AOSM, the matrix B_i^n increases in size by one row and column at a time:

$$B_i^n = \begin{bmatrix} B_i^{n-1} & -(W_i^{n-1})^\top \mathbf{z}_{j\Gamma}^n \\ -(\mathbf{w}_i^n)^\top Z_{j\Gamma}^{n-1} & 1 - (\mathbf{w}_i^n)^\top \mathbf{z}_{j\Gamma}^n \end{bmatrix}.$$

The inverse of B_i^n, $(B_i^n)^{-1}$, can be updated by making use of the Schur complement:

$$\begin{aligned}
(B_i^n)^{-1} =& \begin{bmatrix} (B_i^{n-1})^{-1} & \\ & \end{bmatrix} + \frac{1}{\gamma_n} \begin{bmatrix} \mathbf{s}_n \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{t}_n^\top & 1 \end{bmatrix}, \\
\mathbf{s}_n =& (B_i^{n-1})^{-1} (W_i^{n-1})^\top \mathbf{z}_{j\Gamma}^n, \\
\mathbf{t}_n^\top =& (\mathbf{w}_i^n)^\top Z_{j\Gamma}^{n-1} (B_i^{n-1})^{-1}, \\
\gamma_n =& 1 - (\mathbf{w}_i^n)^\top \mathbf{z}_{j\Gamma}^n - (\mathbf{w}_i^n)^\top Z_{j\Gamma}^{n-1} (B_i^{n-1})^{-1} (W_i^{n-1})^\top \mathbf{z}_{j\Gamma}^n.
\end{aligned}$$

The iterative procedure for computing these inverses is summarized in Algorithm 3.1. Note that forming the inverse through matrix sums and vector outer products is not the most efficient procedure. It is faster to store the inverse as the sequences $\mathbf{s}_n$, $\mathbf{t}_n$ and γ_n and, whenever the product $B_n^{-1}\mathbf{x}$ is needed, to perform it as a series of vector-vector products:

$$B_n^{-1}\mathbf{x} = \sum_{k=1}^{n} \frac{1}{\gamma_k} \begin{bmatrix} \mathbf{s}_k \\ 1 \\ \end{bmatrix} \begin{bmatrix} \mathbf{t}_k^\top & 1 & \end{bmatrix} \mathbf{x}.$$

If done in this manner, then B_n^{-1} can be stored similarly to some factorizations, see below.

$$\begin{array}{|c|c|c|c|}
\hline
\gamma_1 & \mathbf{s}_2 & \multirow{2}{*}{$\mathbf{s}_3$} & \multirow{3}{*}{$\mathbf{s}_4$} \\
\cline{1-2}
\mathbf{t}_2^\top & \gamma_2 & & \\
\cline{1-3}
\mathbf{t}_3^\top & & \gamma_3 & \\
\hline
\multicolumn{2}{c|}{\mathbf{t}_4^\top} & & \gamma_4 \\
\cline{3-4}
\end{array}$$

Algorithm 3.1 Schur shuffle

1: $B_1^{-1} = 1/\gamma_1$
2: **for** $n = 2$ to M **do**
3: Find $\mathbf{s}_n$, $\mathbf{t}_n$ and γ_n
4: Form $B_n^{-1} = \begin{bmatrix} B_{n-1}^{-1} & \\ & \end{bmatrix} + \frac{1}{\gamma_n} \begin{bmatrix} \mathbf{s}_n \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{t}_n^\top & 1 \end{bmatrix}$

Note that equation (3.3) requires only the last column of B_n^{-1}, which is

$$B_n^{-1}\begin{bmatrix}\\ 1\end{bmatrix} = \frac{1}{\gamma_n}\begin{bmatrix}\mathbf{s}_n\\ 1\end{bmatrix}.$$

However, since the entirety of B_{n-1}^{-1} is required to calculate γ_n and $\mathbf{s}_n$, this does not allow us to ignore the calculation of B_n^{-1}.

4. Convergence analysis. We propose that the AOSMs described in subsection 2.3 are Krylov subspace methods. In general, these methods have two components: the construction of a basis of a Krylov subspace, and; an optimization of the solution in this Krylov subspace. We discuss each component in turn.

4.1. Krylov subspaces. We show that the difference vectors lie in Krylov subspaces. We begin by performing the static condensation process to equations (2.2) and (2.3). Firstly, from the first block row in each equation, we have that

$$\mathbf{u}_j^{n+1} = A_{jj}^{-1}\mathbf{f}_j - A_{jj}^{-1}A_{j\Gamma}\mathbf{u}_{j\Gamma}^{n+1}.$$

We plug this into the second block row to produce a fixed point iteration:

$$\begin{aligned}
\left(\hat{A} + E_{i\to j}^{n+1}\right)\mathbf{u}_{j\Gamma}^{n+1} &= \mathbf{f}_\Gamma - A_{\Gamma j}A_{jj}^{-1}\mathbf{f}_j - A_{\Gamma i}A_{ii}^{-1}\mathbf{f}_i + E_{i\to j}^{n+1}\mathbf{u}_{i\Gamma}^{n},\\
&= \mathbf{f}_K + E_{i\to j}^{n+1}\mathbf{u}_{i\Gamma}^{n}, \qquad (4.1)\\
\mathbf{u}_{j\Gamma}^{n+1} &= \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1}\mathbf{f}_K + \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1}E_{i\to j}^{n+1}\left(\hat{A} + E_{j\to i}^{n}\right)^{-1}\mathbf{f}_K\\
&\quad + \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1}E_{i\to j}^{n+1}\left(\hat{A} + E_{j\to i}^{n}\right)^{-1}E_{j\to i}^{n}\mathbf{u}_{j\Gamma}^{n-1}\\
&= \mathbf{f}_{jK}^{n+1} + A_{jK}^{n+1}\mathbf{u}_{j\Gamma}^{n-1}, \qquad (4.2)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{f}_K &:= \mathbf{f}_\Gamma - A_{\Gamma j}A_{jj}^{-1}\mathbf{f}_j - A_{\Gamma i}A_{ii}^{-1}\mathbf{f}_i,\\
\mathbf{f}_{jK}^{n+1} &:= \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1}\mathbf{f}_K + \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1}E_{i\to j}^{n+1}\left(\hat{A} + E_{j\to i}^{n}\right)^{-1}\mathbf{f}_K,\\
A_{jK}^{n+1} &:= \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1}E_{i\to j}^{n+1}\left(\hat{A} + E_{j\to i}^{n}\right)^{-1}E_{j\to i}^{n}.
\end{aligned}$$

The matrix A_{jK}^{n+1} represents the sequential solves of equations (2.2) and (2.3). The vector $\mathbf{f}_{jK}^{n+1}$ is the effect of the right hand side of equation (2.1) on the solution for the j–th subdomain.

Through manipulation of equation (2.1) we find

$$\begin{aligned}
\begin{bmatrix}A_{jj} & A_{j\Gamma}\\ A_{\Gamma j} & A_{\Gamma\Gamma} + T_{i\to j}^{n+1}\end{bmatrix}\begin{bmatrix}\mathbf{u}_j\\ \mathbf{u}_\Gamma\end{bmatrix} &= \begin{bmatrix}\mathbf{f}_j\\ \mathbf{f}_\Gamma\end{bmatrix} - \begin{bmatrix}\\ A_{\Gamma i}\mathbf{u}_i\end{bmatrix} + \begin{bmatrix}\\ T_{i\to j}^{n+1}\mathbf{u}_\Gamma\end{bmatrix},\\
\left(\hat{A} + E_{i\to j}^{n+1}\right)\mathbf{u}_\Gamma &= \mathbf{f}_K + E_{i\to j}^{n+1}\mathbf{u}_\Gamma. \qquad (4.3)
\end{aligned}$$

Rearranging to isolate for $\mathbf{u}_\Gamma$ gives

$$\left(I - \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1}E_{i\to j}^{n+1}\right)\mathbf{u}_\Gamma = \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1}\mathbf{f}_K \ \forall n. \qquad (4.4)$$

Meanwhile, composing equation (4.3) with the same using $E^{n+1}_{j\to i}$ gives

$$(I - A^{n+1}_{jK})\mathbf{u}_\Gamma = \mathbf{f}^{n+1}_{jK} \ \forall n. \tag{4.5}$$

Note that A^n_{jK} is defined for $n \geq 2$. The Krylov subspaces may be represented using A^n_{jK} defined for fixed transmission conditions. We use A^1_{jK} to denote these matrices.

LEMMA 4.1. *Suppose the difference vectors $\mathbf{d}^{n+1}_{j\Gamma}$ are the result of solving equations (2.7) and (2.8) sequentially. Then, for both fixed and adaptive transmission conditions using equation (2.16),*

$$\mathbf{d}^{2k}_{j\Gamma} \in \mathcal{K}_k(A^1_{jK}, \mathbf{d}^2_{j\Gamma}), \quad \mathbf{d}^{2k+1}_{i\Gamma} \in \mathcal{K}_k(A^1_{iK}, \mathbf{d}^3_{i\Gamma}),$$

where

$$A^1_{jK} := \left(\hat{A} + E^1_{i\to j}\right)^{-1} E^1_{i\to j} \left(\hat{A} + E^1_{j\to i}\right)^{-1} E^1_{j\to i}.$$

REMARK 4.2. *While the difference vectors lie in the same Krylov subspaces for both fixed and adaptive transmission conditions, only those for adaptive transmission conditions have been optimized. Those for fixed transmission conditions lack optimization.*

Proof. We begin with fixed transmission conditions, then prove the span of the difference vectors is unchanged by adaptive transmission conditions. For fixed transmission conditions, $\Delta T^n_{i\to j} = 0$ for all n, reducing equations (2.7) and (2.8) to

$$\mathbf{d}^{2k}_{j\Gamma} = \left(\hat{A} + E^1_{i\to j}\right)^{-1} E^1_{i\to j} \mathbf{d}^{2k-1}_{i\Gamma}.$$

Composing this equation for both subdomains results in

$$\mathbf{d}^{2k}_{j\Gamma} = \left(\hat{A} + E^1_{i\to j}\right)^{-1} E^1_{i\to j} \left(\hat{A} + E^1_{j\to i}\right)^{-1} E^1_{j\to i} \mathbf{d}^{2k-2}_{j\Gamma} = A^1_{jK} \mathbf{d}^{2k-2}_{j\Gamma}.$$

This proves the statement of the lemma for fixed transmission conditions.

For adaptive transmission conditions, there are two changes to the difference vectors: addition of $\Delta T^n_{i\to j}$ in equations (2.7) and (2.8), and; use of Algorithm 2.1 to update the transmission conditions. We begin by considering the latter. Algorithm 2.1 uses modified Gram-Schmidt to produce vectors $\mathbf{w}^{2k-1}_i$ from the vectors $\mathbf{d}^{2k-1}_{i\Gamma}$, and $\mathbf{v}^{2k-1}_i$ from $E^{2k-1}_{i\to j}\mathbf{d}^{2k-1}_{i\Gamma}$. Thus, the span of the vectors $\mathbf{w}^{2k-1}_i$ is precisely that of the vectors $\mathbf{d}^{2k-1}_{i\Gamma}$. Moreover, due to Theorem 2.1, the vectors $\mathbf{v}^{2k-1}_i$ lie in the span of $E^1_{i\to j}\mathbf{d}^{2k-1}_{i\Gamma}$.

By equation (3.2), we have a set of vectors $\mathbf{z}^{2k}_{j\Gamma}$ that lie in the span of the vectors $(\hat{A} + E^1_{i\to j})^{-1}\mathbf{v}^{2k-1}_i$. By Theorem 3.2,

$$\begin{aligned}
\mathbf{d}^{2k}_{j\Gamma} &\in \operatorname{span}\left(\left\{\mathbf{z}^{2\ell}_{j\Gamma}\right\}_{\ell=1}^{k}\right) \\
&\in \operatorname{span}\left(\left\{(\hat{A} + E^1_{i\to j})^{-1}\mathbf{v}^{2\ell-1}_i\right\}_{\ell=1}^{k}\right) \\
&\in \operatorname{span}\left(\left\{(\hat{A} + E^1_{i\to j})^{-1}E^1_{i\to j}\mathbf{d}^{2\ell-1}_{i\Gamma}\right\}_{\ell=1}^{k}\right).
\end{aligned}$$

This may then be composed with the same result in the other subdomain, resulting in

$$\begin{aligned}\mathbf{d}_{j\Gamma}^{2k} \in &(\hat{A} + E_{i\to j}^1)^{-1} E_{i\to j}^1 \operatorname{span}\left(\left\{(\hat{A} + E_{j\to i}^1)^{-1} E_{j\to i}^1 \mathbf{d}_{j\Gamma}^{2\ell}\right\}_{\ell=1}^{k-1}\right) \\ &\in \operatorname{span}\left(\left\{A_{jK}^1 \mathbf{d}_{j\Gamma}^{2\ell}\right\}_{\ell=1}^{k-1}\right),\end{aligned}$$

which proves the statement of the lemma for adaptive transmission conditions. ∎

The sequential solves of equations (2.7) and (2.8) indicate multiplicative Schwarz. In this case, one subdomain admits only odd values of n for $\mathbf{d}_{j\Gamma}^{n+1}$ while the other admits only even values of n. For additive Schwarz with fixed transmission conditions, the statement of Theorem 4.1 remains true, as this is simply two instances of multiplicative Schwarz. However, for additive Schwarz with adaptive transmission conditions, the Krylov subspace is augmented.

LEMMA 4.3. *Suppose the difference vectors $\mathbf{d}_{j\Gamma}^{n+1}$ are the result of solving equations (2.7) and (2.8) in parallel with adaptive transmission conditions using equation (2.16). Then*

$$\mathbf{d}_{j\Gamma}^{n+1} \in \mathcal{K}_k(A_{jK}^1, \mathbf{d}_{j\Gamma}^1) + \mathcal{K}_l(A_{jK}^1, \mathbf{d}_{j\Gamma}^2),$$

where $k = \lfloor n/2 \rfloor$ and $l = \lfloor (n-1)/2 \rfloor$.

Proof. We proceed by induction. The first solves of equations (2.7) and (2.8) with the prescribed adaptive transmission conditions result in $\mathbf{d}_{j\Gamma}^2 = (\hat{A} + E_{i\to j}^1)^{-1} E_{i\to j}^1 \mathbf{d}_{i\Gamma}^1$ for both subdomains. This gives the base case, where $n = 1$.

Equation (3.2) and Theorem 3.2 are unchanged by adaptive transmission conditions, and so

$$\begin{aligned}\mathbf{d}_{j\Gamma}^{n+1} &\in \operatorname{span}\left(\left\{(\hat{A} + E_{i\to j}^1)^{-1} E_{i\to j}^1 \mathbf{d}_{i\Gamma}^k\right\}_{k=1}^{n}\right) \\ &\in \operatorname{span}\left(\left\{A_{jK}^1 \mathbf{d}_{j\Gamma}^k\right\}_{k=1}^{n-1}, \mathbf{d}_{j\Gamma}^2\right),\end{aligned}$$

since there is no $\mathbf{d}_{j\Gamma}^0$ such that $\mathbf{d}_{j\Gamma}^2 = A_{jK}^1 \mathbf{d}_{j\Gamma}^0$. By the induction hypothesis, this means

$$\mathbf{d}_{j\Gamma}^{n+1} \in A_{jK}^1 \left(\mathcal{K}_{k-1}(A_{jK}^1, \mathbf{d}_{j\Gamma}^1) + \mathcal{K}_{l-1}(A_{jK}^1, \mathbf{d}_{j\Gamma}^2)\right) + \operatorname{span}(\mathbf{d}_{j\Gamma}^2),$$

which proves the lemma. ∎

4.2. Optimization. Ideally, we seek an update to the solution that lies within the Krylov subspace W_j^{n+1} that minimizes the residual in the standard Euclidean norm. Such an update would be equivalent to GMRES. Sadly, it does not appear that this minimization occurs under the AOSMs described in subsection 2.3. Instead, we find that the following Galerkin condition is imposed.

THEOREM 4.4. *If $\hat{A} + E_{i\to j}^{n+1}$ is invertible, then the update to the solution due to an AOSM is*

$$\mathbf{d}_{j\Gamma}^{n+1} = \left(\hat{A} + E_{i\to j}^n\right)^{-1} E_{i\to j}^n \mathbf{x},$$

where $\mathbf{x} \in \operatorname{span}(W_i^n)$ such that the residual of equation (4.4) applied to $\mathbf{u}_{i\Gamma}^{n-1} + \mathbf{x}$ is orthogonal to $\operatorname{span}(W_i^n)$.

Proof. If $\hat{A} + E_{i\to j}^{n+1}$ is invertible, then by Theorem 3.1

$$\left(\hat{A} + E_{i\to j}^{n} - \mathbf{v}_i^n \left(\mathbf{w}_i^n\right)^\top\right)^{-1} \mathbf{v}_i^n = \frac{\left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} \mathbf{w}_i^n}{\left(\mathbf{w}_i^n\right)^\top \left(I - \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n}\right) \mathbf{w}_i^n},$$

following the same steps as equation (3.3). The denominator is then non-zero by this assumption. From equation (2.18), we may then express $\mathbf{d}_{j\Gamma}^{n+1}$ as

$$\begin{aligned}\mathbf{d}_{j\Gamma}^{n+1} &= \frac{\left(\mathbf{w}_i^n\right)^\top \left(\mathbf{u}_{j\Gamma}^n - \mathbf{u}_{i\Gamma}^{n-1}\right)}{\left(\mathbf{w}_i^n\right)^\top \left(I - \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n}\right) \mathbf{w}_i^n} \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} \mathbf{w}_i^n \\ &= \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} \mathbf{w}_i^n \gamma.\end{aligned}$$

The residual of equation (4.4) applied to $\mathbf{u}_{i\Gamma}^{n-1}$ is

$$\left(\hat{A} + E_{i\to j}^{n}\right)^{-1} \mathbf{f}_K - \left(I - \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n}\right) \mathbf{u}_{i\Gamma}^{n-1} = \mathbf{u}_{j\Gamma}^n - \mathbf{u}_{i\Gamma}^{n-1},$$

using equation (4.1) to retrieve $\mathbf{u}_{j\Gamma}^n$. The Galerkin condition then requires $\mathbf{x} \in \operatorname{span}(W_i^n)$ such that

$$\mathbf{u}_{j\Gamma}^n - \mathbf{u}_{i\Gamma}^{n-1} - \left(I - \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n}\right) \mathbf{x} \perp W_i^n.$$

Denoting $\mathbf{x} = W_i^n \mathbf{y}$ and using $E_{i\to j}^n W_i^{n-1} = 0$ and $(W_i^n)^\top W_i^n = I$, this condition is equivalent to

$$\begin{bmatrix} I & -\left(W_i^{n-1}\right)^\top \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} \mathbf{w}_i^n \\ \mathbf{0}^\top & \left(\mathbf{w}_i^n\right)^\top \left(I - \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n}\right) \mathbf{w}_i^n \end{bmatrix} \mathbf{y} = \left(W_i^n\right)^\top \left(\mathbf{u}_{j\Gamma}^n - \mathbf{u}_{i\Gamma}^{n-1}\right). \tag{4.6}$$

The matrix for this system is invertible by the assumption that $\hat{A} + E_{i\to j}^{n+1}$ is invertible, see above.

Since $E_{i\to j}^n W_i^{n-1} = 0$, we have that

$$\left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} \mathbf{x} = \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} W_i^n \mathbf{y} = \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} \mathbf{w}_i^n y_n.$$

The value of y_n is easily established as γ from equation (4.6). This is then precisely the representation of $\mathbf{d}_{j\Gamma}^{n+1}$ found above, proving the theorem. ∎

This highlights similarites with the full orthogonalization method (FOM) [22, 23] applied to equation (4.4) with implicit restarts [16]. However, the Krylov subspaces described in subsection 4.1 would come from FOM applied to equation (4.5). This creates tension between the two components, though the effect of this tension is unknown.

Note that Theorem 4.4 uses $\mathbf{u}_{i\Gamma}^{n-1}$ to form $\mathbf{d}_{j\Gamma}^{n+1}$, even though at this stage in the algorithm $\mathbf{u}_{i\Gamma}^n$, the result of calculating $\mathbf{d}_{i\Gamma}^n$, is available. The latter does not appear in equation (2.18) and so we are right to ignore it, however, the question lingers as to

what happens to it in the course of the algorithm. Certainly, it appears subsequently when calculating $\mathbf{d}_{j\Gamma}^{n+2}$:

$$\left(\mathbf{w}_i^{n+1}\right)^\top \left(\mathbf{u}_{j\Gamma}^{n+1} - \mathbf{u}_{i\Gamma}^{n}\right) = \left(\mathbf{w}_i^{n+1}\right)^\top \left(\mathbf{u}_{j\Gamma}^{n} + \mathbf{d}_{j\Gamma}^{n+1} - \mathbf{u}_{i\Gamma}^{n-1} - \mathbf{d}_{i\Gamma}^{n}\right),$$

but $\mathbf{w}_i^{n+1}$ is orthogonal to $\mathbf{d}_{i\Gamma}^{n}$ by construction. Indeed, $\mathbf{w}_i^{n+1}$ is orthogonal to all vectors $\mathbf{d}_{i\Gamma}^{k}$ for $k < n+1$, meaning

$$\left(\mathbf{w}_i^{n+1}\right)^\top \left(\mathbf{u}_{j\Gamma}^{n+1} - \mathbf{u}_{i\Gamma}^{n}\right) = \left(\mathbf{w}_i^{n+1}\right)^\top \left(\mathbf{u}_{j\Gamma}^{n+1} - \mathbf{u}_{i\Gamma}^{0}\right).$$

Thus, the vector $\mathbf{d}_{i\Gamma}^{n}$ is not needed for the residual in the i-th subdomain, though it is necessary for the residual in the j-th subdomain.

4.3. Breakdown. Breakdown in standard Krylov subspace methods are events that halt progress. We now explore possible breakdown scenarios in AOSMs and their effect on the solution. Algorithm 2.2 and Algorithm 2.3 can break down or stagnate either in the call to Algorithm 2.1 or in solving equation (2.18).

Algorithm 2.1 breaks down when the input vectors are linearly dependent. As a result, no new $\mathbf{w}_i^n$ can be calculated. In the course of the AOSMs, this occurs when $\mathbf{d}_{i\Gamma}^{n} \in \operatorname{span}(W_i^{n-1})$. The following proposition shows this breakdown is 'lucky', in the sense that the solution at this point is exact.

PROPOSITION 4.5. *If $\mathbf{d}_{i\Gamma}^{n} \in \operatorname{span}(W_i^{n-1})$, then the update to $\mathbf{u}_{j\Gamma}^{n-3}$ due to Algorithm 2.2 (altAOSM) eliminates the residual of equation (4.5).*

Proof. If $\mathbf{d}_{i\Gamma}^{n} \in \operatorname{span}(W_i^{n-1})$, then $E_{i\to j}^{n}\mathbf{d}_{i\Gamma}^{n} = 0$. Thus,

$$\begin{aligned} A_{jK}^{n} W_j^{n-1} &= \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} \left(\hat{A} + E_{j\to i}^{n-1}\right)^{-1} E_{j\to i}^{n-1} W_j^{n-1} \\ &= \left(\hat{A} + E_{i\to j}^{n}\right)^{-1} E_{i\to j}^{n} \begin{bmatrix} 0 & \gamma \mathbf{d}_{i\Gamma}^{n} \end{bmatrix} = 0, \end{aligned}$$

for some $\gamma \in \mathbb{R}$, see equation (2.18). Note that, since Algorithm 2.2 updates the transmission conditions every second iteration, $A_{jK}^{n} = A_{jK}^{n-1}$, while $A_{jK}^{n+1} \neq A_{jK}^{n}$. Since the vector $\mathbf{d}_{j\Gamma}^{n-1} \in \operatorname{span}(W_j^{n-1})$, $A_{jK}^{n-1}\mathbf{d}_{j\Gamma}^{n-1} = 0$. The residual of equation (4.5) applied to $\mathbf{u}_{j\Gamma}^{n-3} + \mathbf{d}_{j\Gamma}^{n-1}$ is then, simplifying using equation (4.2),

$$\mathbf{f}_{jK}^{n-1} - \left(I - A_{jK}^{n-1}\right)\left(\mathbf{u}_{j\Gamma}^{n-3} + \mathbf{d}_{j\Gamma}^{n-1}\right) = \mathbf{u}_{j\Gamma}^{n-1} - \mathbf{u}_{j\Gamma}^{n-3} - \mathbf{d}_{j\Gamma}^{n-1} = 0,$$

by definition of $\mathbf{d}_{j\Gamma}^{n-1}$. This update then eliminates the residual. ∎

REMARK 4.6. *Algorithm 2.3 will only break down in this way if both $\mathbf{d}_{i\Gamma}^{n}, \mathbf{d}_{i\Gamma}^{n-1} \in \operatorname{span}(W_i^{n-2})$. The proof is adapted by replacing $\mathbf{d}_{j\Gamma}^{n-1}$ wherever it appears by $\mathbf{d}_{j\Gamma}^{n-1} + \mathbf{d}_{j\Gamma}^{n-2}$, adjusting indices as necessary.*

Equation (2.18) cannot be solved when the matrix

$$I - \left(W_i^{n}\right)^\top Z_{j\Gamma}^{n} = \left(W_i^{n}\right)^\top \left(I - \left(\hat{A} + E_{i\to j}^{1}\right)^{-1} E_{i\to j}^{1}\right) W_i^{n}$$

is singular. In this case, no update $\mathbf{d}_{j\Gamma}^{n+1}$ can be calculated, see equation (3.3). The nullspace of this matrix is the set of Ritz vectors satisfying

$$\left(W_i^{n}\right)^\top \left(\hat{A} + E_{i\to j}^{1}\right)^{-1} E_{i\to j}^{1} W_i^{n}\mathbf{y} = \mathbf{y},$$

that is, with Ritz value equal to 1. Ritz values lie within the numerical range, also known as the field of values [2], meaning they are bounded by the numerical radius. This stagnation can then be avoided if it is possible to choose $E^1_{i\to j}$ such that the numerical range does not contain 1. We discuss this possibility.

The numerical range of a matrix contains the convex hull of its eigenvalues. The number 1 is not an eigenvalue of the matrix in question, as such vectors imply a non-trivial solution to equation (2.1) with $\mathbf{f} = 0$, as may be seen in equation (4.4). The numerical radius is bounded by the largest singular value of the matrix, and so a sufficient condition to avoid hard breakdown is

$$\left\| \left(\hat{A} + E^1_{i\to j} \right)^{-1} E^1_{i\to j} \right\|_{op} < 1,$$

where $\|\cdot\|_{op}$ represents the operator norm induced by the 2-norm.

Note that, since W^n_i span a Krylov subspace, the Ritz values are akin to Arnoldi-Ritz values. However, the Krylov subspace is developed from a different matrix, A^1_{jK}, meaning this is not a precise characterization.

The AOSMs stagnate if $(\mathbf{w}^n_i)^\top (\mathbf{u}^n_{j\Gamma} - \mathbf{u}^{n-1}_{i\Gamma}) = 0$, as this results in a null right hand side in equation (2.18). This can occur if the residual of equation (4.4) lies entirely within $\text{span}(W^n_i)$, which would indicate the exact solution is found by Theorem 4.4. In general, this stagnation does not appear to be equivalent to convergence.

In other Krylov subspace methods, wise choices of seed vectors of the Krylov subspaces, here denoted $\mathbf{w}^1_i$, based on the initial residual are enough to give equivalence between this type of stagnation and convergence. However, in Algorithm 2.2 this vector and the residual are both chosen through $\mathbf{u}^0_{i\Gamma}$. There is more choice in Algorithm 2.3, since one chooses both $\mathbf{u}^0_{i\Gamma}$ and $\mathbf{u}^0_{j\Gamma}$, but this will still leave one subdomain without choice of seed vector. This is an example of the tension between the two Krylov subspace components discussed earlier.

5. Numerical examples.

5.1. Comparison of the different versions of AOSMs. Subsection 2.3 describes two broad categories of AOSMs, altAOSM which is applied to multiplicative Schwarz and paraAOSM which is applied to additive Schwarz. For each of these categories, we have three versions: a standard version which solves equations (2.2) and (2.3); a corrector version which solves equation (2.18), and; the corrector version coupled with the Woodbury matrix identity, which solves equation (3.3) in both subdomains. All three versions produce theoretically equivalent solution and difference vectors. However, corrector versions are often more numerically stable and the Woodbury matrix identity can improve efficiency. We will compare all three versions to verify equivalence and test numerical stability. To confirm Theorem 4.4, we also compare a fourth version which directly applies the Galerkin condition in place of equation (2.18).

We choose as a benchmark Poisson's equation in 2D,

$$\begin{cases} \Delta u(x,y) = f(x,y), & (x,y) \in \Omega = [-1,1] \times [-1,1], \\ u(x,y) = g(x,y), & (x,y) \in \partial\Omega. \end{cases} \tag{5.1}$$

We use an evenly spaced grid of N points, such that there are $\sqrt{N}$ points in the x-direction and $\sqrt{N}$ points in the y-direction. The grid spacing is then $h = 2/(\sqrt{N} - 1)$. The operator Δ may be represented using a 5-point finite difference stencil.The

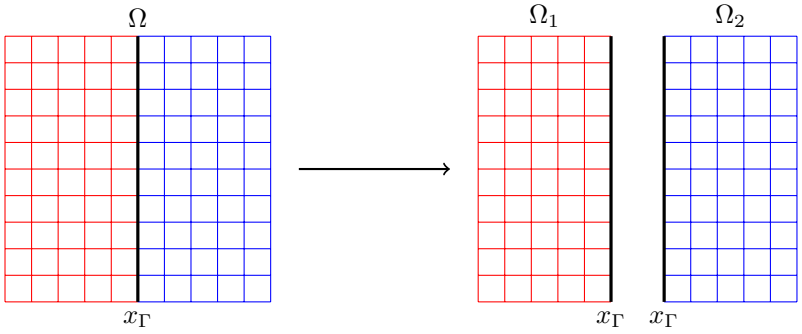

Fig. 5.1: Splitting of one domain into two non-overlapping subdomains. This is the algebraic splitting for both Dirichlet and Robin transmission conditions, but is the physical splitting only for Robin transmission conditions.

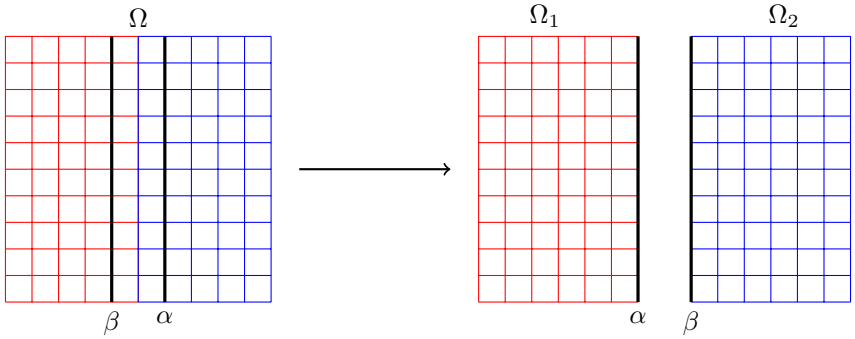

Fig. 5.2: Splitting of one domain into two overlapping subdomains. When using Dirichlet boundary conditions, this is the equivalent physical splitting.

Dirichlet boundary conditions may be implemented either by directly replacing the values on the stencils, or by using appropriate rows of the identity matrix. We opt for the former [25], which is equivalent to choosing $T^1_{i\to j} = 0$ for both subdomains.

We split the domain into two subdomains, Ω_1 and Ω_2, along a given value of $x = x_\Gamma$, see Figure 5.1. This split is primarily algebraic, meaning only the variables on the interface of the domain are repeated. However, when using Dirichlet transmission conditions implemented as described, the boundary of the physical subdomains is one step away from this interface, see Figure 5.2. This gives a physical overlap of $2h$, where h is the size of the elements. Using Robin transmission conditions, the physical subdomains are as pictured in Figure 5.1 and the method is non-overlapping.

As has been noted in subsection 2.1, the zeroth order OSM for this problem is $T^1_{1\to 2} = T^1_{2\to 1} = -\frac{1}{2}A_{\Gamma\Gamma} + pI$ [5], with $p = -\pi/h^{3/2}$ when using the convention that the diagonal entries of the matrix are $-4/h^2$. If using the convention that the diagonal entries are 4, then multiply through by $-h^2$. We compare the three versions of altAOSM and the comparable GMRES in solving equation (5.1) with $N = 10,000$, $N_1 = 4900$, $N_2 = 5000$, and $M = 100$, as well as $f = g = 1$. Initial guesses are chosen

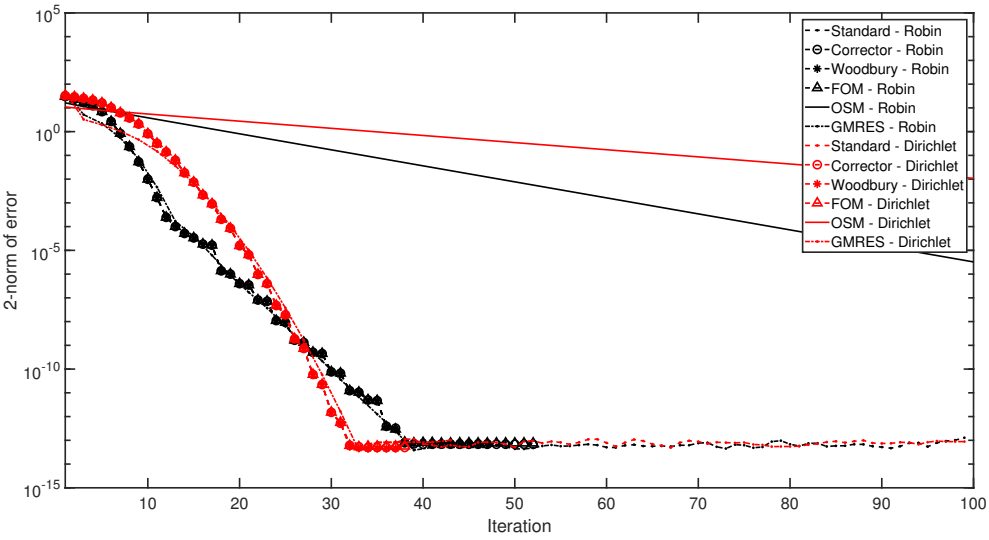

Fig. 5.3: Comparison of altAOSM versions, including standard, corrector, Woodbury and FOM, in solving equation (5.1) with $N = 10,000$, $M = 100$, using both Robin and Dirichlet boundary conditions. Also shown is the OSM with and without GMRES for this problem.

to be zero. Error is measured as the Euclidean distance between the solution obtained by each method and the control solution obtained by solving the full system. Note this then does not account for any approximation error, only for the error resulting from using a Schwarz method to solve the system.

The results are seen in Figure 5.3. We can see that all four versions of altAOSM, for both types of boundary conditions, are numerically equivalent. There are minor stability differences at the level of saturation, which may cause issues for more complicated problems.

The corresponding OSM is shown for benchmark purposes. For OSM to converge properly, its initial guess is chosen at random so that it contains modes of all frequencies [5]. We see that altAOSM significantly outperforms this more basic method. We also see that indeed the Dirichlet boundary conditions are significantly slower than the optimized Robin conditions. When accelerating the OSM with GMRES, convergence is comparable to altAOSM.

Robin boundary conditions, which have been optimized according to known OSM standards [25], provide faster convergence in altAOSM for the first number of iterations. After this, convergence slows. The method remains fast compared with the OSM and achieves the same level of saturation as altAOSM starting from Dirichlet boundary conditions.

Figure 5.4 shows the same trends hold true of paraAOSM, namely numerical equivalence between the four versions with minor discrepancies in stability at saturation. GMRES-accelerated OSM no longer has comparable convergence to paraAOSM. Robin conditions offer improved convergence for OSM but not for paraAOSM. In comparing altAOSM with paraAOSM, we note that paraAOSM achieves faster convergence due to the larger Krylov subspace explored at each iteration. Note also that each iteration of paraAOSM requires roughly twice as much computation as an

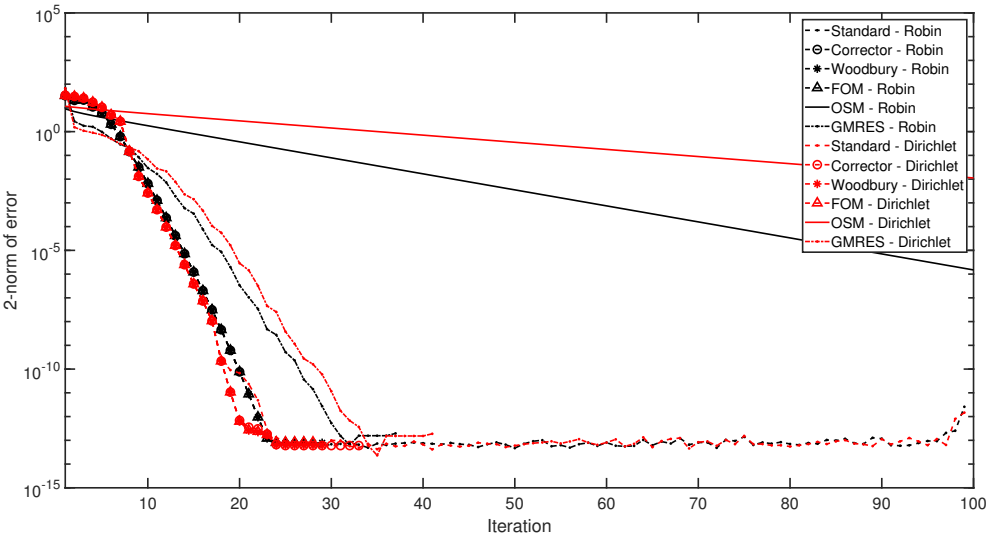

Fig. 5.4: Comparison of paraAOSM versions, including standard, corrector, Woodbury and GMRES, in solving equation (5.1) with $N = 10,000$, $M = 100$, using both Robin and Dirichlet boundary conditions. Also shown is the OSM for this problem.

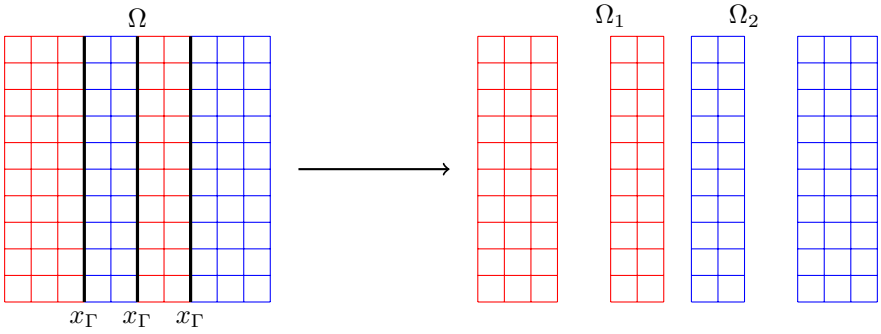

Fig. 5.5: Splitting of one domain into two non-overlapping red-black subdomains.

iteration of altAOSM, though paraAOSM is fully parallelizable, meaning with two processors the time to compute each iteration should be the same.

5.2. Red-black decompositions and multiple subdomains. Let us now consider other possible splittings of the subdomain, in particular a red-black decomposition. In such a decomposition, the domain is split into many pieces, such as strips or rectangular domains for 2D problems. These pieces are then grouped into two subdomains, such that each subdomain alternates pieces. This decomposition takes its name from the checkerboard shape that results when splitting rectilinear grids into smaller squares. Figure 5.5 shows the red-black decomposition we will use in this example, which splits our square domain into strips.

This decomposition is similar to a strip-wise decomposition into multiple sub-

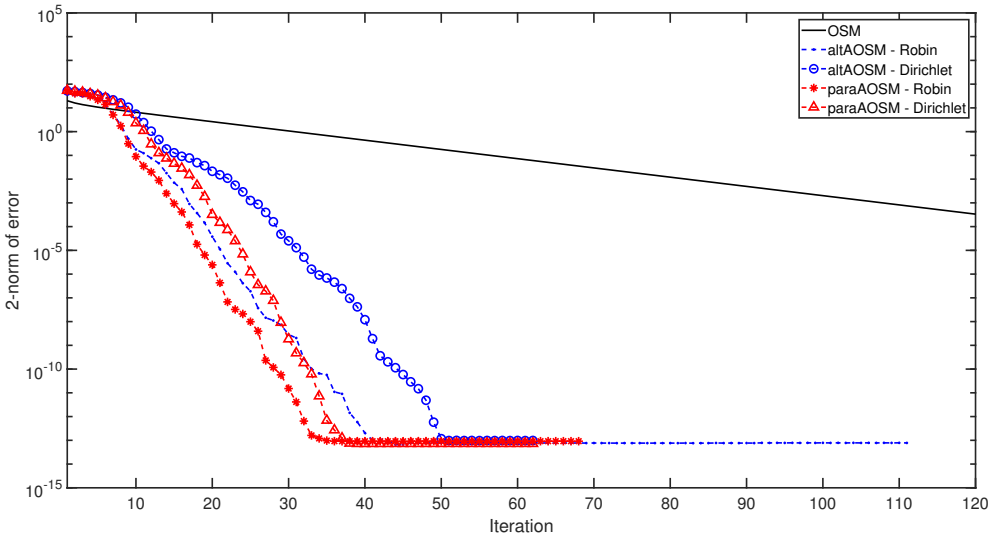

Fig. 5.6: Convergence of OSM, altAOSM, and paraAOSM in solving equation (5.1) with a red-black decomposition.

domains. Indeed, each of the blocks defined in equation (2.1) are block diagonal, in particular A_{11} and A_{22} which permits the full parallelization of their inversions. However, the Schur complements are full, as are the adaptive transmission conditions $T_{i\to j}^{n+1}$ that result from the AOSM. This prevents a complete decomposition into multiple subdomains with this version of the AOSM.

For this example, we apply five methods: the corresponding OSM and both altAOSM and paraAOSM with optimized Robin and Dirichlet conditions as initial transmission conditions. The domain, with $N = 10,000$, is split into four strips such that each subdomain stretches over all 100 values of y. The subdomains alternate possession of the values of x, such that the first 29 belong exclusively to the first subdomain, then an interface, then 19 in the second subdomain, then an interface, then 19 in the first, then an interface, then 30 in the second. In this way, $N_1 = 4800$, $N_2 = 4900$ and $M = 300$.

The results are presented in Figure 5.6. All AOSMs converge to saturation well before the artificial limit of M iterations, which represents roughly the amount of computation required to compute the Schur complement. As before, we see that optimized initial transmission conditions do not necessarily support faster convergence than Dirichlet conditions. The paraAOSM performs slightly better than altAOSM, and both significantly outperform the OSM.

We repeat this example with a checkerboard red-black decomposition, as seen in Figure 5.7. The splitting uses the same division of the values of x, then repeats this division over the values of y. In this way, $N_1 = 4704$, $N_2 = 4705$ and $M = 591$. There are nine crosspoints which complicate the OSM for this problem. By setting the transmission conditions to zero at these points, we retrieve a convergent OSM.

Again, A_{11} and A_{22} are block diagonal, permitting parallelization of their inversions, while the Schur complements and transmission conditions are full. Results of this example are presented in Figure 5.8 and show similarities to those of Figure 5.6. Note that M is roughly twice as large for this example, resulting in roughly twice as

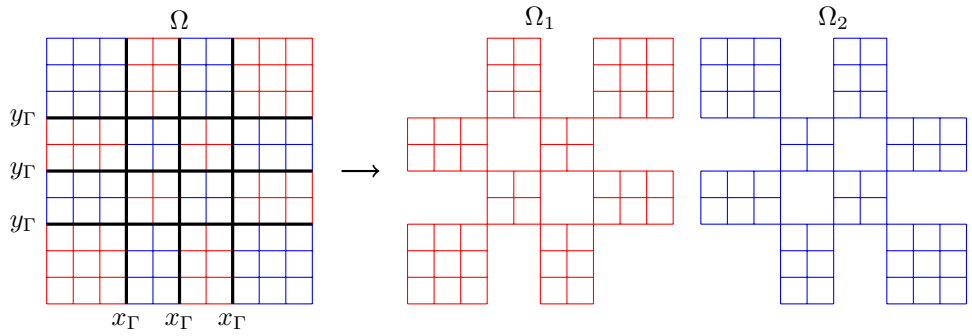

Fig. 5.7: Splitting of one domain into two non-overlapping checkerboard subdomains.

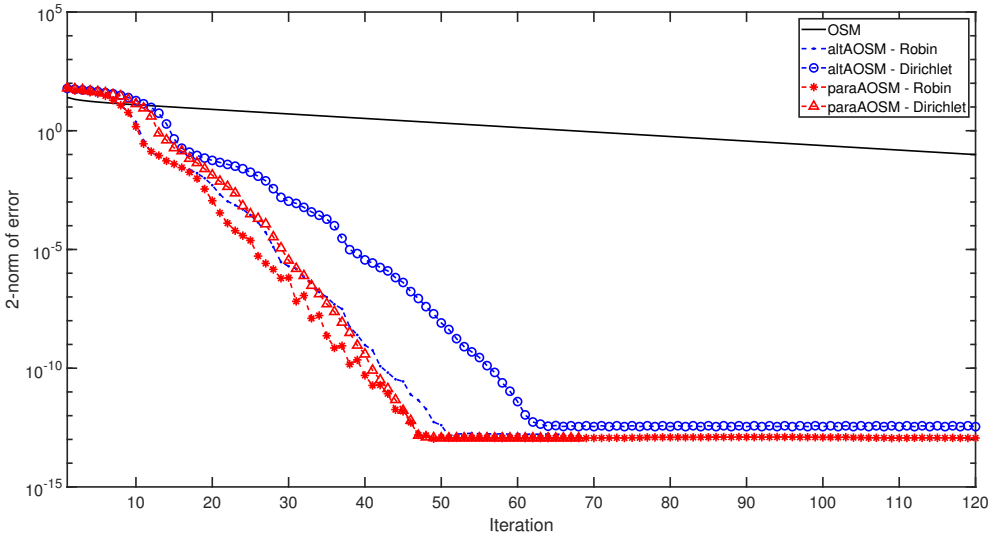

Fig. 5.8: Convergence of OSM, altAOSM, and paraAOSM in solving equation (5.1) with a checkerboard red-black decomposition.

many iterations required to reach saturation of the error.

As stated, the transmission conditions $T_{i\to j}^{n+1}$ are full, a problem when generalizing the AOSM to multiple subdomains. Figure 5.9 shows the contours of one of these transmission conditions, specifically $T_{1\to 2}^{n+1}$ resulting from the altAOSM. The left of this figure shows this result for the strip-wise decomposition. It is clear from this figure that this matrix has a 3×3 block structure, corresponding to the three interfaces of the decomposition. The right of the figure uses the checkerboard decomposition and shows a significantly more complicated block structure. The matrix $T_{i\to j}^{n+1}$ would then be better resolved by considering each of its blocks separately. This requires significant modification to the AOSM and would likely permit a complete decomposition into multiple subdomains.

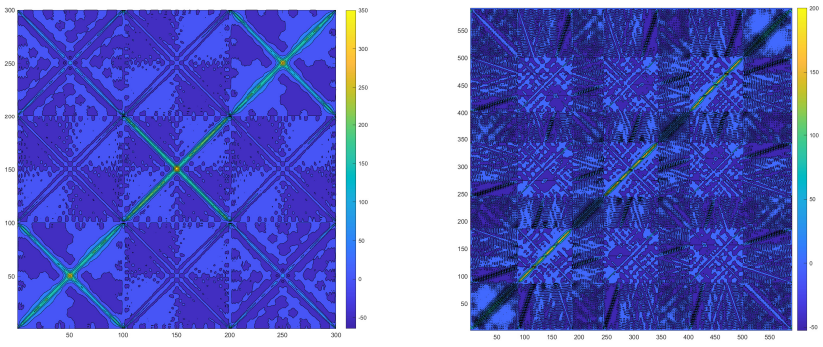

Fig. 5.9: Contours of the matrix $T_{1\to 2}^{n+1}$ that results from altAOSM applied to the (left) strip-wise and (right) checkerboard red-black decompositions.

5.3. Helmholtz equation. We next apply the two AOSMs to the Helmholtz equation, to show its effectiveness on non-SPD problems. We set

$$\begin{cases} \Delta u(x,y) + k^2 u(x,y) = 0, & (x,y) \in \Omega = [-1,1] \times [-1,1], \\ u(x,y) = 1, & (x,y) \in \partial\Omega. \end{cases} \tag{5.2}$$

We use the same discretization and subdomains as previously for the Laplace operator. That is, we use an evenly spaced grid of N points with grid spacing $h = 2/(\sqrt{N}-1)$ in both the x and y-directions. The operator Δ is represented with a 5-point finite difference stencil. The domain is split algebraically into two subdomains along a given value of $x = x_\Gamma$, see Figure 5.1, such that $N_1 = 4900$, $N_2 = 5000$, and $M = 100$. The value of k is set to $2\pi/10h$, resulting in 10 grid points per wavelength. Dirichlet transmission conditions are used initially, thus there is a physical overlap of $2h$.

Note that equation (5.2) is more delicate than the one with complex Robin conditions on the outer boundary. For the latter, one can optimize the parameters in the transmission conditions to obtain good convergence for all frequencies [11]. This is not possible for the Dirichlet problem because of reflected waves [14]. We continue with Dirichlet boundary conditions to show the effectiveness of the AOSMs over optimized Schwarz methods.

We test this problem using altAOSM and paraAOSM with initial Dirichlet transmission conditions. For comparison, we also use multiplicative Schwarz with acceleration by GMRES and additive Schwarz with and without GMRES.

Results are presented in Figure 5.10. Convergence for the four Krylov methods, which are the two AOSMs and the Schwarz methods accelerated with GMRES, follow similar convergent trajectories. The altAOSM appears to suffer a stability issue which slows down its convergence in comparison to its multiplicative counterpart, while the paraAOSM and its additive counterpart closely mirror one another. The standard additive Schwarz method does not converge, which previous research tells us is due to the boundary conditions in the problem [11, 14].

5.4. Heat equation. AOSMs may be particularly suited for time-dependent problems, given that the transmission conditions resulting from each time step can be

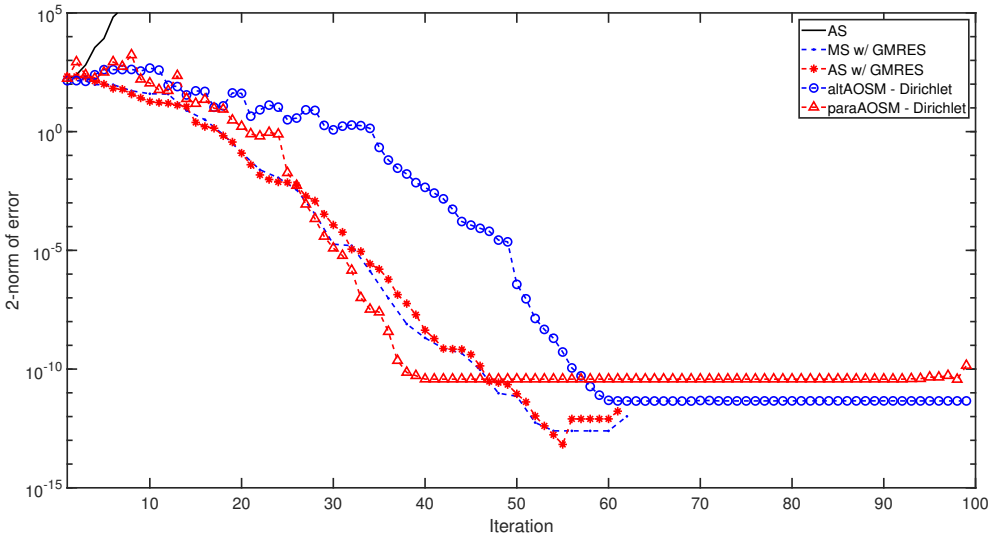

Fig. 5.10: Convergence of additive Schwarz, multiplicative Schwarz with GMRES, additive Schwarz with GMRES, altAOSM and paraAOSM in solving equation (5.2). The AOSMs and GMRES-accelerated Schwarz methods have similar convergence curves, while additive Schwarz diverges.

stored. This will give AOSMs on subsequent time steps a head start, or possibly even allow the time steps to be completed in two iterations, if the transmission conditions are sufficiently resolved.

To test this, we examine the heat equation, in the form

$$\begin{cases} u_t(x,y,t) = \Delta u(x,y,t), & (x,y) \in \Omega = [-1,1]\times[-1,1],\ t \in [0,T] \\ u(x,y,0) = u_0(x,y), & (x,y) \in \Omega, \\ u(x,y,t) = g(x,y), & (x,y) \in \partial\Omega,\ t \in [0,T]. \end{cases} \tag{5.3}$$

In space, we again use a finite difference formulation so that the same matrix A from subsection 5.1 can be used to represent the Laplacian. In time, we use a backward Euler integration scheme, such that

$$\frac{u_{n+1} - u_n}{\Delta t} = A u_{n+1}.$$

This gives a system to solve at each time step of the form

$$(I - \Delta t A) u_{n+1} = u_n. \tag{5.4}$$

As in subsection 5.1, we split the domain into two non-overlapping subdomains, of the form of Figure 5.1. We choose $u_0(x,y) = 1$ and $g(x,y) = 0$. We take again $N = 10,000$, $N_1 = 4900$, $N_2 = 5000$, and $M = 100$. The time step is chosen as $\Delta t = 0.01$. We use optimized Robin boundary conditions as initial transmission conditions, which uses $T^1_{1\to 2} = T^1_{2\to 1} = -\frac{1}{2}A_{\Gamma\Gamma} + pI$ where

$$p = \sqrt{\frac{\pi}{h^3}}\left(\frac{\pi^2}{4} + \frac{1}{\Delta t}\right)^{1/4} \Delta t,$$

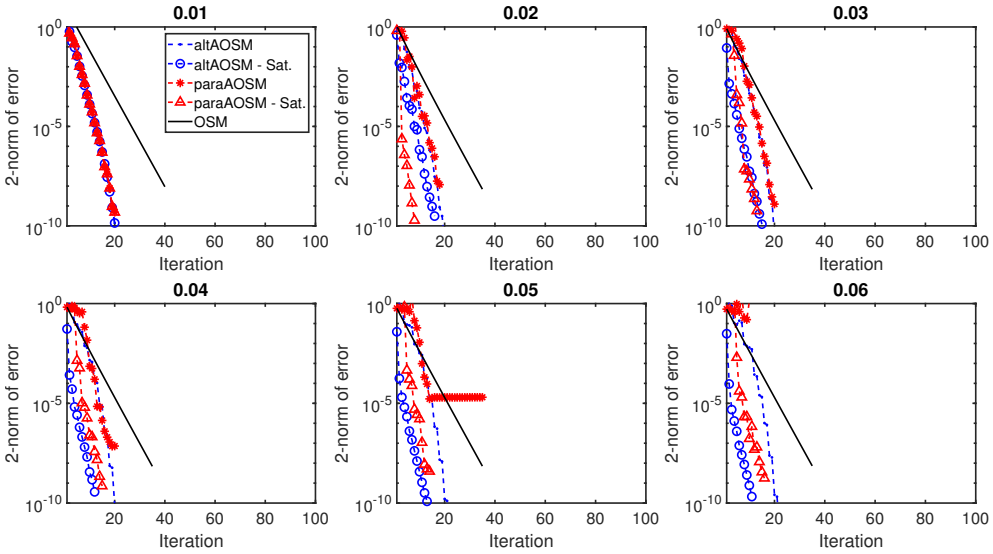

Fig. 5.11: Convergence of OSM and AOSMs at the first six time steps of equation (5.4). AOSMs that use updated transmission conditions at each time step are labelled 'Sat'.

which may be retrieved from [25]. Again, if the diagonal entries of A are 4 instead of $-4/h^2$, then multiply these transmission conditions by $-h^2$. We seek convergence to a tolerance of 10^{-8}.

We test five methods. For each type of AOSM, altAOSM and paraAOSM, we test one version using the initial transmissions throughout and a second version which uses the adaptive transmission conditions found from the previous time step. We refer to this second version as 'saturated'. The first time step is then identical for each pair. These four methods are then compared against the OSM for this problem.

We are not concerned with the discretisation error between the physical solution to equation (5.3) and that found by solving equation (5.4). Instead, we focus on the approximation error between solving equation (5.4) directly, which we will call the control solution, and solving the equation using an AOSM, specifically alt-AOSM.

The first six steps of each method are presented in Figure 5.11. The OSM maintains steady convergence at each step, though it is slow compared with the AOSMs. The stopping criteria for the AOSMs causes over-convergence, allowing the error to fall several orders of magnitude below what is required.

The convergence rate of the AOSMs is similar across these time steps. As the transmission conditions are adapted, this convergence is sped up slightly. While altAOSM maintains good accuracy throughout these time steps, the paraAOSM declines in stability. At $t = 0.05$, we see that the unsaturated paraAOSM begins to diverge from the true solution.

We count the number of iterations each of the five methods requires to achieve tolerance for twelve steps. The results are presented in Table 5.1. Both OSM and altAOSM consistently achieve their tolerance. The paraAOSMs are unable to achieve their tolerance as time marches on.

From these results, we can conclude that paraAOSM has stability issues if left

Table 5.1: Iteration counts for each time step of equation (5.4). A dash indicates the method failed to converge to the desired tolerance.

t	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
altAOSM	20	20	20	20	22	22	22	24	30
altAOSM - Sat.	20	16	15	12	13	11	11	11	11
paraAOSM	20	18	20	20	-	-	-	-	-
paraAOSM - Sat.	20	8	13	15	14	16	13	14	15
OSM	40	35	35	35	35	35	35	35	35

t	0.10	0.11	0.12
altAOSM	26	-	-
altAOSM - Sat.	8	8	7
paraAOSM	-	-	-
paraAOSM - Sat.	-	-	-
OSM	35	35	35

unattended. This is most likely due to the manner in which the augmented Krylov subspace is constructed which, unlike in standard GMRES, is formed as the sum of two separate Krylov subspaces. A specialized version of modified Gram-Schmidt or stabilizing techniques from Krylov subspace method literature may improve the method to working order. One can also consider using two non-interacting instances of altAOSM in the same way that additive Schwarz is two non-interacting instances of multiplicative Schwarz. This will not improve convergence rates or accuracy, but it will produce a solution on every subdomain at each iteration.

We can also conclude that re-using the adapted transmission conditions at subsequent time steps leads to better results. In both versions of AOSM, this reduces the number of iterations required and improves stability. Since these adaptations to the transmission conditions are low rank updates, it is possible that they can become oversaturated once the updates reach full rank.

One can also improve the manner in which the transmission conditions are reused. In the present example, the recycled transmission conditions are treated as new matrices $T^1_{i\to j}$. It may be beneficial to instead treat them as adaptations of the original transmission conditions from the first step, thereby preserving the orthogonality of the vectors $\mathbf{w}^n_j$ and the nullspace of the matrices $E^n_{i\to j}$.

5.5. Heterogeneous elliptic example on an unstructured grid. The previous examples considered are relatively easy for any method to compute. Grids have been rectilinear and PDEs have been tractable. Let us now consider a more challenging PDE to make sure the success of the AOSMs is not restricted to trivial computations. To that end, we consider the PDE

$$\begin{cases} -\nabla\left(\alpha(x,y)\cdot\nabla u(x,y)\right) = f(x,y), & (x,y)\in\Omega=[0,1]\times[0,1], \\ u(x,y) = g(x,y), & (x,y)\in\partial\Omega. \end{cases} \tag{5.5}$$

We choose $\alpha(x,y)=1$ except along three thin channels in the domain, where $\alpha(x,y)=1000$. This example is adapted from [10].

We use finite element method software, MEF++, to construct the domain and linear system, resulting in an unstructured grid with $N=937$ points. The value of

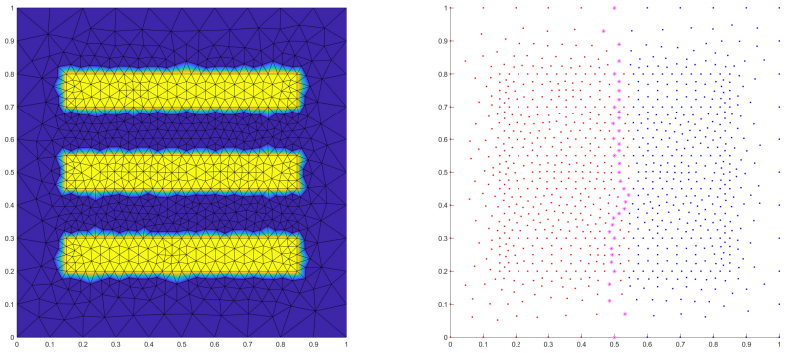

Fig. 5.12: (Left) Values of $\alpha(x, y)$ over the unstructured grid generated by MEF++ for equation (5.5). Blue represents the value of 1, while yellow represents the value of 1000. (Right) Subdivision of the domain into two subdomains. Red and blue nodes represent the points exclusive to one of the two subdomains, while pink stars represent points shared by the two subdomains.

$\alpha(x, y)$ is given for this grid in Figure 5.12, left. The three channels are intended to strongly link the two subdomains.

We subdivide the domain manually such that the interface is roughly vertical, see Figure 5.12, right. 37 points are chosen for the interface, 487 for the first subdomain, and 413 for the second. The optimized Robin transmission conditions are approximately $T^1_{i\to j} = -0.5A_{\Gamma\Gamma} + pK$, where K is a diagonal matrix containing the values of $\alpha(x, y)$ along the interface and p is a constant not specifically prescribed [17]. We find that $p = 0.67$ is roughly optimal. For true optimized Robin transmission conditions, both the contribution of $A_{\Gamma\Gamma}$ and K must be scaled to account for the difference in sizes of the elements, leading to non-constant p and $T^1_{1\to 2} \neq T^1_{2\to 1}$. An alternative to manual selection is the reverse Cuthill-McKee algorithm [15] applied to the matrix A generated by the FEM software, which minimizes the bandwidth of the matrix, and then selecting a suitable block. This does not guarantee a straight interface, which may require adapting the OSM.

We seek convergence to 10^{-8}. We start with zero boundary conditions and a unit source. Figure 5.13 shows the convergence curves of the methods tested: the corresponding OSM; altAOSM and paraAOSM with Dirichlet initial transmission conditions, and; altAOSM and paraAOSM with Dirichlet initial transmisson conditions. Due to the high disparity in values of $\alpha(x, y)$ throughout the domain, the OSM fails to converge. The AOSMs also struggle, but manage to reach tolerance within the artificial limit of M iterations. Dirichlet boundary conditions work best as initial transmission conditions, which is at odds with expectations but consistent with previous results.

Note that if the channels are extended to the boundaries of the domain, the problem is significantly easier to solve numerically. The OSM becomes a convergent method, while the AOSMs converge to the tolerance in roughly half the number of

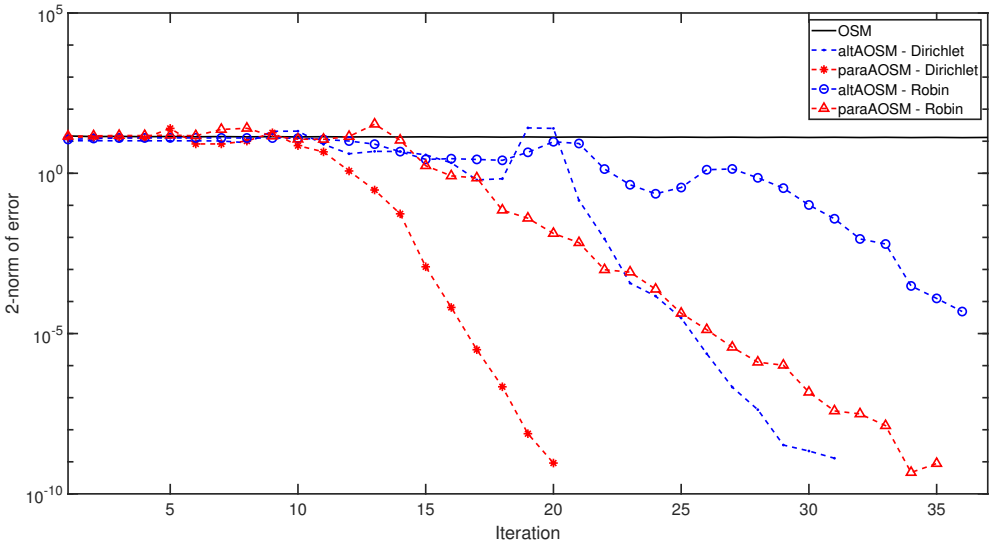

Fig. 5.13: Convergence of OSM and AOSMs in solving equation (5.5) with $g(x, y) = 0$ and $f(x, y) = 1$.

iterations.

We are interested in whether we can indeed re-use the transmission conditions that result from the AOSMs. Let us consider a second right hand side with zero boundary conditions and a source function

$$f(x, y) = (x - 1/2)^2(y - 1/2)^2.$$

There are several options on how to re-use these transmission conditions. We test eight variants: For each of the four AOSMs tested with the first right hand side, we use the resulting adapted transmission conditions directly in an OSM, thus keeping them fixed at their current update, and as initial transmission conditions for the same methods, thereby implicitly restarting them.

The results are presented in Figure 5.14. On the left, we see the OSMs with recycled transmission conditions converge linearly, with those transmission conditions adapted from Dirichlet conditions permitting fastest convergence. The original OSM is included for comparison. As before, it fails to converge.

On the right of Figur Figure 5.14, we have the 'implicitly restarted' AOSMs. Each now converges significantly faster than previously. Again, Dirichlet conditions appear to be the better choice of initial transmission conditions, as their results show faster convergence and greater stability.

Comparing between these methods, it appears to be unnecessary to continue to adapt transmission conditions once a good set has been found, as the OSM is able to converge in comparative number of iterations as the 'implicitly restarted' methods.

6. Conclusions and future work. We have presented two methods, Algorithm 2.2 and Algorithm 2.3, for adaptively optimizing transmission conditions of a Schwarz method for any nonsingular algebraic system. The adaptations arise from an algebraic decomposition of the system and can therefore be employed with minimal knowledge of the continuous problem.

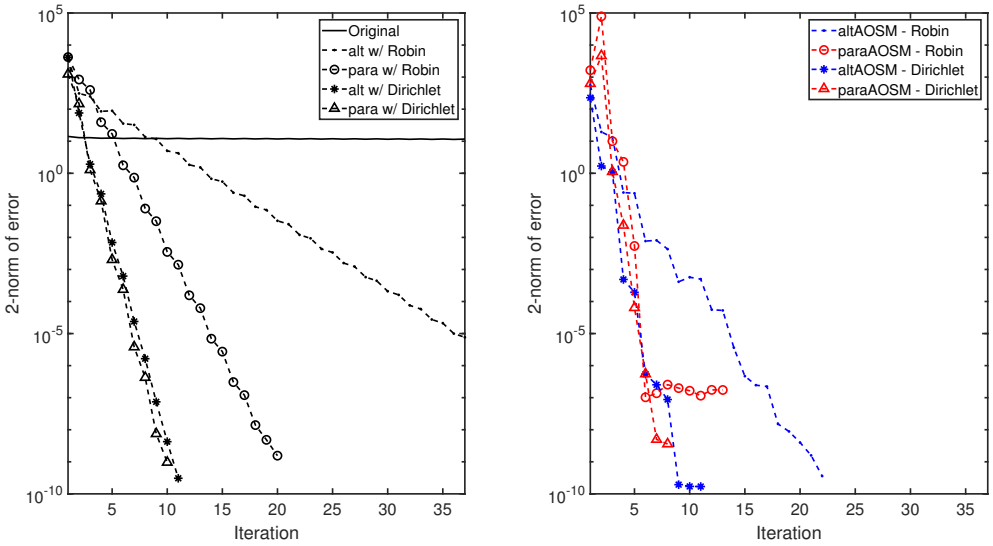

Fig. 5.14: Convergence of OSM (left) and AOSMs (right) in solving equation (5.5) with $g(x, y) = 0$ and $f(x, y)$ a quadratic function and recycled transmission conditions.

The adaptations are rank one updates intended to expand the nullspace of the matrices $E^1_{i\to j}$, the difference between the initial transmission conditions and the relevant Schur complements, which represent the optimal transmission conditions, see subsection 2.2. The core principle is to find a decomposition of $E^1_{i\to j}$ that can be added to the transmission conditions.

We have shown that, with these adaptive transmission conditions, the iterates lie within Krylov subspaces, or the sum of Krylov subspaces in the case of paraAOSM, and satisfy a Galerkin condition, see section 4. We discussed three cases of breakdown, one which was equivalent to convergence, one which could be avoided, and one which remains a possibility.

Our numerical examples have shown the AOSMs are applicable broadly, and may be especially useful for time dependent problems or situations with multiple right hand sides, see section 5.

The next major step is to generalize the AOSMs to an arbitrary number of domains. Subsection 5.2 details options similar to this generalization that can be implemented within a two-subdomain framework. These show that a full generalization to multiple subdomains is significantly more complicated than the AOSM presented here.

Several modifications are possible and worthy of investigating. Is there a choice of adaptation that allows replacement of the Galerkin condition with stronger optimization in the Krylov subspace? What is the cause of the stability issues of paraAOSM? Can multipreconditioning be employed to extract adapted transmission conditions faster? How can these methods be applied to more complicated decompositions?

7. Acknowledgements. F. Kwok acknowledges support from the National Science and Engineering Research Council of Canada (RGPIN- 2021-02595). The work described in this paper is partially supported by a grant from the ANR/RGC joint research scheme sponsored by the Research Grants Council of the Hong Kong Special

Administrative Region, China and the French National Research Agency (Project no.s ANR-19-CE46-0013 and A-CityU203/19).

C. McCoid acknowledges funding provided through the CRM-Laval Postdoctoral Fellowship, offered jointly by the Centre de recherches mathématiques and the Université Laval.

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] R. L. Carden and M. Embree, *Ritz value localization for non-Hermitian matrices*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 1320–1338.
[3] V. Dolean, P. Jolivet, and F. Nataf, *An introduction to domain decomposition methods: algorithms, theory, and parallel implementation*, SIAM, 2015.
[4] B. Engquist and A. Majda, *Absorbing boundary conditions for numerical simulation of waves*, Proceedings of the National Academy of Sciences, 74 (1977), pp. 1765–1766.
[5] M. J. Gander, *Optimized Schwarz methods*, SIAM Journal on Numerical Analysis, 44 (2006), pp. 699–731.
[6] M. J. Gander, L. Halpern, F. Nataf, et al., *Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation*, in Eleventh international Conference of Domain Decomposition Methods. ddm. org, 1999.
[7] M. J. Gander, L. Halpern, F. Nataf, et al., *Optimized Schwarz methods*, in Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan, 2001, pp. 15–28.
[8] M. J. Gander and F. Kwok, *Chladni figures and the Tacoma bridge: motivating PDE eigenvalue problems via vibrating plates*, SIAM Review, 54 (2012), pp. 573–596.
[9] M. J. Gander, S. Loisel, and D. B. Szyld, *An optimal block iterative method and preconditioner for banded matrices with applications to PDEs on irregular domains*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 653–680.
[10] M. J. Gander and A. Loneland, *SHEM: An optimal coarse space for RAS and its multiscale approximation*, in Domain decomposition methods in science and engineering XXIII, Springer, 2017, pp. 313–321.
[11] M. J. Gander, F. Magoules, and F. Nataf, *Optimized Schwarz methods without overlap for the helmholtz equation*, SIAM Journal on Scientific Computing, 24 (2002), pp. 38–60.
[12] M. J. Gander, R. Masson, and T. Vanzan, *A numerical algorithm based on probing to find optimized transmission conditions*, in Domain Decomposition Methods in Science and Engineering XXVI, Springer, 2023, pp. 597–605.
[13] M. J. Gander and M. Outrata, *Optimized Schwarz methods with data-sparse transmission conditions*, in Domain Decomposition Methods in Science and Engineering XXVI, Springer, 2023, pp. 471–478.
[14] M. J. Gander and H. Zhang, *Schwarz methods by domain truncation*, Acta Numerica, 31 (2022), pp. 1–134.
[15] A. George and J. W. Liu, *Computer solution of large sparse positive definite*, Prentice Hall Professional Technical Reference, 1981.
[16] G. H. Golub and C. F. Van Loan, *Matrix computations*, JHU press, 4th ed., 2013.
[17] Y. Gu and F. Kwok, *On the choice of Robin parameters for the optimized Schwarz method for domains with non-conforming heterogeneities*, Journal of Scientific Computing, 89 (2021), p. 5.
[18] R. J. Guyan, *Reduction of stiffness and mass matrices*, AIAA journal, 3 (1965), pp. 380–380.
[19] R. L. Higdon, *Numerical absorbing boundary conditions for the wave equation*, Mathematics of computation, 49 (1987), pp. 65–90.
[20] E. H. Moore, *On the reciprocal of the general algebraic matrix*, Bull. Am. Math. Soc., 26 (1920), pp. 394–395.
[21] R. Penrose, *A generalized inverse for matrices*, in Mathematical proceedings of the Cambridge philosophical society, vol. 51, Cambridge University Press, 1955, pp. 406–413.
[22] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Mathematics of computation, 37 (1981), pp. 105–126.
[23] Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003.
[24] J. Sherman, *Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix*, Annals of mathematical statistics, 20 (1949), p. 621.

[25] A. St-Cyr, M. J. Gander, and S. J. Thomas, *Optimized multiplicative, additive, and restricted additive Schwarz preconditioning*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2402–2425.
[26] L. N. Trefethen and L. Halpern, *Well-posedness of one-way wave equations and absorbing boundary conditions*, Mathematics of computation, 47 (1986), pp. 421–435.
[27] M. Woodbury, *Inverting modified matrices. memorandum rep. 42*, Statistical Research Group, Princeton University, (1950).