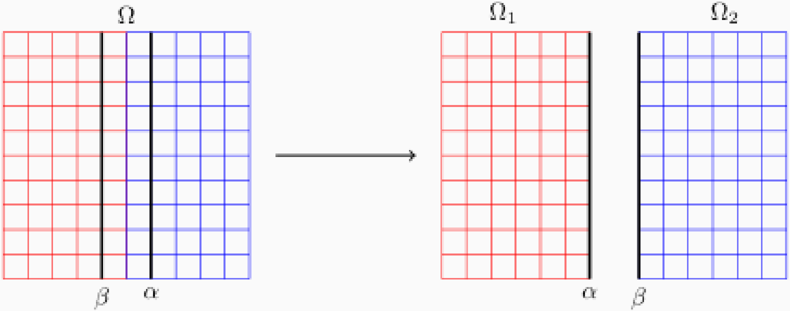
Adaptive optimized Schwarz methods

Conor McCoid

January 24th, 2025

McMaster University

Algebraic Schwarz



Algebraic Schwarz

Consider the following algebraic system with block tridiagonal matrix A:

$$A\boldsymbol{u} = \begin{bmatrix} A_{11} & A_{1\Gamma} & \\ A_{\Gamma 2} & A_{\Gamma\Gamma} & A_{\Gamma 1} \\ & A_{2\Gamma} & A_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_\Gamma \\ \boldsymbol{u}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_\Gamma \\ \boldsymbol{f}_2 \end{bmatrix} = \boldsymbol{f}.$$

The solution is divided into two subdomains: Ω_1 contains the variables in $\boldsymbol{u}_1$ and $\boldsymbol{u}_\Gamma$, and Ω_2 contains the variables in $\boldsymbol{u}_2$ and $\boldsymbol{u}_\Gamma$. The variables $\boldsymbol{u}_\Gamma$ are then shared between the two subdomains. We will refer to this portion as the **interface** between the two subdomains.

The idea is to find local solutions on each subdomain, a process which is easier to accomplish than solving the system as a whole. To find a global solution, local solutions are found iteratively, with

Algebraic Schwarz

There is a physical interpretation of this algebraic splitting, the more commonly seen Schwarz methods. Consider the differential equation

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

Splitting this up using a Schwarz method results in the following subproblems that are solved iteratively:

$$\begin{cases} \Delta u_1^n(x,y) = f(x,y), & (x,y) \in \Omega_1 = [-1,\alpha] \times [-1,1], \\ u_1^n(\alpha,y) = u_2^{n-1}(\alpha,y), \end{cases}$$

$$\begin{cases} \Delta u_2^n(x,y) = f(x,y), & (x,y) \in \Omega_2 = [\beta,1] \times [-1,1], \\ u_2^n(\beta,y) = u_1^{n-1}(\beta,y). \end{cases}$$

The **overlap**, physical equivalent of the interface, is now the region where $\beta < x < \alpha$. Note that Dirichlet boundary conditions on this

Algebraic Schwarz

The two subdomains in algebraic Schwarz are

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

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

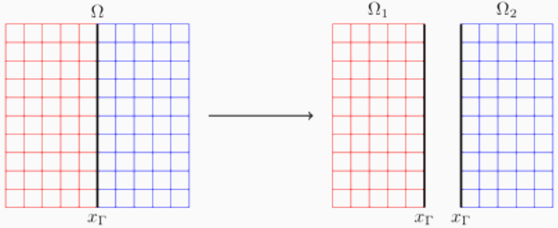
for any choice of matrices $T_{1\to 2}$ and $T_{2\to 1}$ (that does not make these systems singular).

Each choice of matrices T corresponds to a choice of boundary conditions on the interface, called **transmission conditions**, since they transmit information between subdomains.

Algebraic Schwarz

For example, choosing $T = 0$ is equivalent to Dirichlet transmission conditions (in most cases). Note in this case, the Dirichlet information is taken from *outside* the interface.

There is thus a disconnect between the physical overlap and the algebraic interface. In fact, the same interface can represent both overlapping and non-overlapping subdomains, depending on T.



Algebraic Schwarz

Dirichlet transmission conditions generally provide slow convergence. Optimized Schwarz methods use higher order transmission conditions with parameters to improve this convergence. Easiest among these is Robin transmission conditions:

$$\frac{\partial u_1^n}{\partial x} - p u_1^n(0,y) = \frac{\partial u_2^{n-1}}{\partial x} - p u_2^{n-1}(0,y),$$

where one can find an optimal p. This usually takes upfront Fourier analysis or similar methods, but produces significant speed-ups.

One can even go higher order, such as with tangential transmission conditions:

$$\frac{\partial u_1^n}{\partial x} - p u_1^n(0,y) + q\frac{\partial u_1^n}{\partial y} = \frac{\partial u_2^{n-1}}{\partial x} - p u_2^{n-1}(0,y) + q\frac{\partial u_2^n}{\partial y},$$

now giving parameters p and q to tweak.

One can continue going to higher and higher order, with increasingly non-local information transmitted, until one has as many parameters to adjust as there are variables in the interface. At the limit, one arrives at the optimal transmission conditions, **absorbing boundary conditions (ABCs)**, which are traditionally used to ensure incoming waves do not reflect at the boundary. There are numerous methods for approximating ABCs, such as Padé approximants and perfectly matched layers, each with their own benefits and drawbacks.

Algebraically, ABCs correspond to replacing the T matrices with the Schur complements:

$$T_{i \to j} \to S_{i \to j} := -A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma}.$$

With these, algebraic Schwarz with two subdomains converges in

Algebraic Schwarz

Of course, if one has the Schur complements, algebraic Schwarz becomes a direct method. But the Schur complements are expensive to calculate, and in general dense. In the time it takes to calculate the Schur complements, one can run roughly the same number of subdomain solves as there are variables in the interface.

This puts an artificial limit on the number of iterations of algebraic Schwarz to run. Let M be the number of variables in the interface. If algebraic Schwarz does not converge in M iterations, we would have been better off computing the Schur complements.

Adaptive Transmission Conditions

We aim to construct black box approximations to these Schur complements / ABCs using vectors produced in the course of the algebraic Schwarz method. These approximations will be updated at each iteration of the method. The transmission conditions will then be **adaptive**.

This requires reformulating the original algebraic systems with fixed T matrices to ones with adaptive T matrices:

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

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

Now $T_{i\to j}^{n+1}$ represents the transmission conditions used at step $n+1$.

Adaptive Transmission Conditions

How do we extract updates to the transmission conditions? Let us reformulate the systems again, into a corrector form that works on the difference between successive iterates:

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

Adaptive Transmission Conditions

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

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

Adaptive Transmission Conditions

The first line of blocks in each of these systems lets us write

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

Combined with the second line of blocks, the systems reduce to

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

Adaptive Transmission Conditions

Notice the difference between the T matrices and the Schur complements in these systems. If we represent this difference as

$$E_{i\to j}^{n+1} := T_{i\to j}^{n+1} - S_{i\to j},$$

and write $\hat{A} := A_{\Gamma\Gamma} + S_{1\to 2} + S_{2\to 1}$, then these systems become

$$\left(\hat{A} + E_{i\to j}^{n+1}\right) \boldsymbol{d}_{j\Gamma}^{n+1} = E_{i\to j}^{n+1} \boldsymbol{d}_{i\Gamma}^{n} - \Delta T_{i\to j}^{n} \left(\boldsymbol{u}_{j\Gamma}^{n} - \boldsymbol{u}_{i\Gamma}^{n-1}\right).$$

The matrix $E_{i\to j}^{n+1}$ is as expensive to calculate as the Schur complement, meaning this system is not practical. However, it has immense theoretical value.

Adaptive Transmission Conditions

We focus on the matrix-vector product $E_{i\to j}^{n+1}\boldsymbol{d}_{i\Gamma}^{n}$. Tracing backwards, this is equal to

$$E_{i\to j}^{n+1}\boldsymbol{d}_{i\Gamma}^{n} = -A_{\Gamma i}\boldsymbol{d}_{i}^{n} + T_{i\to j}^{n+1}\boldsymbol{d}_{i\Gamma}^{n}.$$

Thus, we can (and must) compute this matrix-vector product in the course of the method.

This gives us a sequence of vector pairs, $(\boldsymbol{d}_{i\Gamma}^{n}, E_{i\to j}^{n+1}\boldsymbol{d}_{i\Gamma}^{n})$. Let's use this sequence to construct an approximation to $E_{i\to j}^{n+1}$. If we have an approximation to $E_{i\to j}^{n+1}$, we can subtract it from $T_{i\to j}^{n+1}$ to get an approximation of $S_{i\to j}^{n+1}$.

Adaptive Transmission Conditions

Let $\boldsymbol{y} = E\boldsymbol{d}$. From one vector pair we can extract a rank one approximation of E:

$$E \approx \frac{\boldsymbol{y}\boldsymbol{d}^\top}{\|\boldsymbol{d}\|_2^2}.$$

This approximation is only accurate when multiplying by vectors parallel to $\boldsymbol{d}$.

Each vector pair gives such a rank one approximation, but the sum of these approximations is only an approximation of E if the vectors $\boldsymbol{d}$ are orthogonal, which is not true in general. To fix this, we apply **modified Gram-Schmidt** to the set of vectors $\boldsymbol{d}$, resulting in the vectors $\boldsymbol{w}$. To get the vectors $\boldsymbol{v} = E\boldsymbol{w}$, a commenserate process must be done on the vectors $\boldsymbol{y}$ (NOT MGS!).

Adaptive Transmission Conditions

Algorithm 1 Iterative action approximation

$[V_n, W_n] = \mathrm{IAA}\left(\{\boldsymbol{x}_k\}_{k=1}^{n}, E\right)$

Inputs: $\{\boldsymbol{x}_k\}_{k=1}^{n} \subset \mathbb{R}^M$, $E \in \mathbb{R}^{M\times M}$

$\alpha_1 := 1/\left\|\boldsymbol{x}_1\right\|$, $\boldsymbol{w}_1 := \alpha_1 \boldsymbol{x}_1$, $\boldsymbol{v}_1 := \alpha_1 E \boldsymbol{x}_1$

for $k = 2 : n$ **do**

  $\boldsymbol{w}_k := \boldsymbol{x}_k$, $\boldsymbol{v}_k := E\boldsymbol{x}_k$

  for $i = 1 : k-1$ **do**

    $h \leftarrow \langle \boldsymbol{w}_i, \boldsymbol{w}_k \rangle$, $\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k - h\boldsymbol{w}_i$

    $\boldsymbol{v}_k \leftarrow \boldsymbol{v}_k - h\boldsymbol{v}_i$

  end for

  $\alpha_k := 1/\left\|\boldsymbol{w}_k\right\|$, $\boldsymbol{w}_k \leftarrow \alpha_k \boldsymbol{w}_k$, $\boldsymbol{v}_k \leftarrow \alpha_k \boldsymbol{v}_k$

end for

Outputs: $W_n = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_n \end{bmatrix}$, $V_n = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_n \end{bmatrix}$

$V_n W_n^\top \approx E$

Adaptive Transmission Conditions

Lemma

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

$$
\begin{aligned}
E^{k+1} =& E(I - W_k W_k^\top), \\
\boldsymbol{v}_k =& E \boldsymbol{w}_k = E^k \boldsymbol{w}_k
\end{aligned}
$$

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

Adaptive Transmission Conditions

For our adaptive transmission conditions, we want

$$[V_i^n, W_i^n] = \mathrm{IAA}\left(\left\{\boldsymbol{d}_{i\Gamma}^k\right\}_{k=1}^n, E_{i\to j}^1\right),$$

$$\text{where } E_{i\to j}^1 \boldsymbol{d}_{i\Gamma}^k = -A_{\Gamma i}\boldsymbol{d}_i^k + T_{i\to j}^1 \boldsymbol{d}_{i\Gamma}^k.$$

Each time we find a new $\boldsymbol{d}_{i\Gamma}^k$, with corresponding $\boldsymbol{d}_i^k$, we can compute a new $\boldsymbol{v}_i^k$ and $\boldsymbol{w}_i^k$. This gives us a rank one update to the transmission conditions:

$$\Delta T_{i\to j}^n = -\boldsymbol{v}_i^k \left(\boldsymbol{w}_i^k\right)^\top.$$

This choice of ΔT eliminates the product $E_{i\to j}^{n+1}\boldsymbol{d}_{i\Gamma}^n$ from the earlier systems. We must now solve

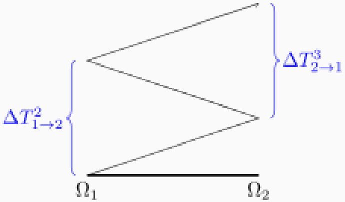
$$\begin{bmatrix} A_{jj} & A_{j\Gamma} \\ A_{\Gamma j} & A_{\Gamma\Gamma} + T_{i\to j}^{n+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_j^{n+1} \\ \boldsymbol{d}_{j\Gamma}^{n+1} \end{bmatrix} = \left(\boldsymbol{w}_i^n\right)^\top \left(\boldsymbol{u}_{j\Gamma}^n - \boldsymbol{u}_{i\Gamma}^{n-1}\right)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

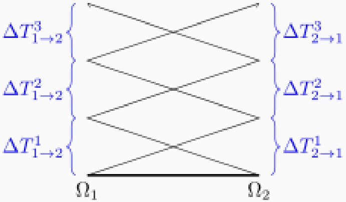
Algorithm 2 altAOSM: AOSM applied to multiplicative Schwarz

Start with initial transmission conditions $T^1_{1\to 2}$ and $T^1_{2\to 1}$
Make initial guess $\boldsymbol{u}^0_{1\Gamma}$
Calculate $\boldsymbol{u}^0_1 = A^{-1}_{11}(\boldsymbol{f}_1 - A_{1\Gamma}\boldsymbol{u}^0_{1\Gamma})$
Solve for $\boldsymbol{u}^1_2$ and $\boldsymbol{u}^1_{2\Gamma}$
Solve for $\boldsymbol{u}^2_1$ and $\boldsymbol{u}^2_{1\Gamma}$
Calculate $\boldsymbol{d}^2_{1\Gamma} = \boldsymbol{u}^2_{1\Gamma} - \boldsymbol{u}^0_{1\Gamma}$ and $\boldsymbol{d}^2_1$ and set $n = 2$
while $\|\boldsymbol{d}^n_{1\Gamma}\| + \left\|\boldsymbol{d}^{n-1}_{2\Gamma}\right\| \geq tol$ **do**
    for $i = 1:2$ **do**
        Run an iteration of IAA
        Set $\Delta T^n_{i\to j} = -\boldsymbol{v}^n_i(\boldsymbol{w}^n_i)^\top$
        Solve for $\boldsymbol{d}^{n+1}_{j\Gamma}$ and $\boldsymbol{d}^{n+1}_j$
        $\boldsymbol{u}^{n+1}_j := \boldsymbol{u}^{n-1}_j + \boldsymbol{d}^{n+1}_j$, $\boldsymbol{u}^{n+1}_{j\Gamma} := \boldsymbol{u}^{n-1}_{j\Gamma} + \boldsymbol{d}^{n+1}_{j\Gamma}$
        $n \leftarrow n + 1$
    end for
end while
Output: $\boldsymbol{u} = [\boldsymbol{u}^n_1\ ;\ (\boldsymbol{u}^n_{1\Gamma} + \boldsymbol{u}^{n-1}_{2\Gamma})/2\ ;\ \boldsymbol{u}^{n-1}_2]$

Algorithm 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 $\boldsymbol{u}^0_{1\Gamma}$ and $\boldsymbol{u}^0_{2\Gamma}$
Calculate $\boldsymbol{u}^0_1 = A^{-1}_{11}(\boldsymbol{f}_1 - A_{1\Gamma}\boldsymbol{u}^0_{1\Gamma})$ and equivalently $\boldsymbol{u}^0_2$
Solve both subdomains for $\boldsymbol{u}^1_1$, $\boldsymbol{u}^1_{1\Gamma}$, $\boldsymbol{u}^1_2$ and $\boldsymbol{u}^1_{2\Gamma}$
Calculate $\boldsymbol{d}^1_{1\Gamma}$, $\boldsymbol{d}^1_1$, $\boldsymbol{d}^1_{2\Gamma}$ and $\boldsymbol{d}^1_2$ and set $n = 1$
while $\|\boldsymbol{d}^n_{1\Gamma}\| + \|\boldsymbol{d}^n_{2\Gamma}\| \geq tol$ **do**
    for $i = 1 : 2$ **do**
        Run an iteration of IAA
        Set $\Delta T^n_{i\to j} = -\boldsymbol{v}^n_i(\boldsymbol{w}^n_i)^\top$
        Solve for $\boldsymbol{d}^{n+1}_{j\Gamma}$ and $\boldsymbol{d}^{n+1}_j$
        $\boldsymbol{u}^{n+1}_j := \boldsymbol{u}^n_j + \boldsymbol{d}^{n+1}_j$, $\boldsymbol{u}^{n+1}_{j\Gamma} := \boldsymbol{u}^n_{j\Gamma} + \boldsymbol{d}^{n+1}_{j\Gamma}$
    end for
    $n \leftarrow n + 1$
end while
Output: $\boldsymbol{u} = [\boldsymbol{u}^n_1 \; ; \; (\boldsymbol{u}^n_{1\Gamma} + \boldsymbol{u}^n_{2\Gamma})/2 \; ; \; \boldsymbol{u}^n_2]$


$\Delta T^2_{1\to 2}$
$\Delta T^3_{2\to 1}$
Ω_1
Ω_2


$\Delta T^3_{1\to 2}$
$\Delta T^2_{1\to 2}$
$\Delta T^1_{1\to 2}$
$\Delta T^3_{2\to 1}$
$\Delta T^2_{2\to 1}$
$\Delta T^1_{2\to 1}$
Ω_1
Ω_2

Woodbury matrix identity

Since the matrices in the linear systems are updated at each iteration, one cannot simply re-use factorizations of these systems. To salvage any factorizations of the initial systems, one can employ the **Woodbury matrix identity**.

Proposition (Woodbury matrix identity)

If $\tilde{\boldsymbol{u}}$ is the solution to $A\tilde{\boldsymbol{u}} = \boldsymbol{b}$, $A \in \mathbb{R}^{N \times N}$, $\boldsymbol{b} \in \mathbb{R}^N$, then

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

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

Woodbury matrix identity

Applying this to the systems we must solve at every iteration, we instead solve

$$\begin{bmatrix} A_{jj} & A_{j\Gamma} \\ A_{\Gamma j} & A_{\Gamma\Gamma} + T^1_{i\to j} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}^n_j \\ \boldsymbol{z}^n_{j\Gamma} \end{bmatrix} = \begin{bmatrix} \\ \boldsymbol{v}^n_i \end{bmatrix},$$

with some factorization, then update our vectors using

$$\begin{bmatrix} \boldsymbol{d}^{n+1}_j \\ \boldsymbol{d}^{n+1}_{j\Gamma} \end{bmatrix} = (\boldsymbol{w}^n_i)^\top \left(\boldsymbol{u}^n_{j\Gamma} - \boldsymbol{u}^{n-1}_{i\Gamma}\right) \Bigg(\begin{bmatrix} \boldsymbol{z}^n_j \\ \boldsymbol{z}^n_{j\Gamma} \end{bmatrix} + \begin{bmatrix} Z^n_j \\ Z^n_{j\Gamma} \end{bmatrix} \left(I - (W^n_i)^\top Z^n_{j\Gamma}\right)^{-1} (W^n_i)^\top \boldsymbol{z}^n_{j\Gamma} \Bigg),$$

where $Z^n_{j\Gamma} = \begin{bmatrix} \boldsymbol{z}^1_{j\Gamma} & \dots & \boldsymbol{z}^n_{j\Gamma} \end{bmatrix}$, etc.

Woodbury matrix identity

Because the right hand side appears in the Woodbury matrix identity, we can perform some simplifications to the previous equation:

$$
\begin{aligned}
&\begin{bmatrix} \boldsymbol{z}_j^n \\ \boldsymbol{z}_{j\Gamma}^n \end{bmatrix} + \begin{bmatrix} Z_j^n \\ Z_{j\Gamma}^n \end{bmatrix} \left(I - (W_i^n)^\top Z_{j\Gamma}^n \right)^{-1} (W_i^n)^\top \boldsymbol{z}_{j\Gamma}^n \\
&= \begin{bmatrix} Z_j^n \\ Z_{j\Gamma}^n \end{bmatrix} \left(I + \left(I - (W_i^n)^\top Z_{j\Gamma}^n \right)^{-1} (W_i^n)^\top Z_{j\Gamma}^n \right) \begin{bmatrix} \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} Z_j^n \\ Z_{j\Gamma}^n \end{bmatrix} \left(I - (W_i^n)^\top Z_{j\Gamma}^n \right)^{-1} \begin{bmatrix} \\ 1 \end{bmatrix}.
\end{aligned}
$$

Krylov subspaces

We can show that the vectors $\boldsymbol{d}_{i\Gamma}^{k}$ lie within Krylov subspaces.

First, if $\Delta T = 0$ (fixed transmission conditions), then

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

Composing this with the same equation for the other subdomain gives

$$\boldsymbol{d}_{j\Gamma}^{n+1} = \left(\hat{A} + E_{i\to j}^{1}\right)^{-1} E_{i\to j}^{1} \left(\hat{A} + E_{j\to i}^{1}\right)^{-1} E_{j\to i}^{1} \boldsymbol{d}_{j\Gamma}^{n-1} = A_{jK}^{1} \boldsymbol{d}_{j\Gamma}^{n-1}.$$

Then clearly $\boldsymbol{d}_{j\Gamma}^{n+1} \in \mathcal{K}_n(A_{jK}^{1}, \boldsymbol{d}_{j\Gamma}^{1})$.

Krylov subspaces

Second, for our adaptive transmission conditions,

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

Composing this with the same result for the other subdomain results in

$$\boldsymbol{d}_{j\Gamma}^{n+1} \in \operatorname{span}\left(\left\{A_{jK}^{1}\boldsymbol{d}_{j\Gamma}^{k}\right\}\right) = \mathcal{K}_n(A_{jK}^{1}, \boldsymbol{d}_{j\Gamma}^{1}).$$

For the paraAOSM, the Krylov subspaces are augmented:

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

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

Optimization of the solution

The ideal update to the solution lies within the Krylov subspace span(W_i^n) and minimizes the residual of

$$(I - A_{jK}^1)\boldsymbol{u}_\Gamma = \boldsymbol{f}_{jK}^1$$

in the Euclidean norm. (A_{jK}^1 is the matrix that produces the Krylov subspace, while $\boldsymbol{f}_{jK}^1$ is an appropriate right hand side constructed from $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ and $\boldsymbol{f}_\Gamma$.) This would give results equivalent to GMRES.

This can be accomplished by saving the Hessenberg matrices produced as part of the IAA and performing an additional (smaller) system solve at each iteration. It requires more computation and more communication between the subdomains.

Optimization of the solution

Without these extra steps, the AOSMs as described perform their own optimization, a Galerkin condition.

Theorem

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

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

where $\boldsymbol{x} \in \operatorname{span}(W_i^n)$ such that the residual of

$$\begin{aligned} &\left(I - \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1} E_{i\to j}^{n+1}\right) \boldsymbol{u}_\Gamma \\ &= \left(\hat{A} + E_{i\to j}^{n+1}\right)^{-1} \left(\boldsymbol{f}_\Gamma - A_{\Gamma j} A_{jj}^{-1} \boldsymbol{f}_j - A_{\Gamma i} A_{ii}^{-1} \boldsymbol{f}_i\right) \end{aligned}$$

applied to $\boldsymbol{u}_{i\Gamma}^{n-1} + \boldsymbol{x}$ is orthogonal to $\operatorname{span}(W_i^n)$.

Optimization of the solution

Proof: Apply the Woodbury matrix identity with only the latest rank one update:

$$
\begin{aligned}
&\left(\hat{A} + E^n_{i\to j} - \boldsymbol{v}^n_i \left(\boldsymbol{w}^n_i\right)^\top\right)^{-1} \boldsymbol{v}^n_i \\
&= \left(\hat{A} + E^n_{i\to j}\right)^{-1} \boldsymbol{v}^n_i \Bigg(I + \\
&\quad \left(I - \left(\boldsymbol{w}^n_i\right)^\top \left(\hat{A} + E^n_{i\to j}\right)^{-1} \boldsymbol{v}^n_i\right)^{-1} \left(\boldsymbol{w}^n_i\right)^\top \left(\hat{A} + E^n_{i\to j}\right)^{-1} \boldsymbol{v}^n_i \Bigg) \\
&= \left(\hat{A} + E^n_{i\to j}\right)^{-1} \boldsymbol{v}^n_i \left(I - \left(\boldsymbol{w}^n_i\right)^\top \left(\hat{A} + E^n_{i\to j}\right)^{-1} \boldsymbol{v}^n_i\right)^{-1} \\
&= \frac{\left(\hat{A} + E^n_{i\to j}\right)^{-1} E^n_{i\to j} \boldsymbol{w}^n_i}{\left(\boldsymbol{w}^n_i\right)^\top \left(I - \left(\hat{A} + E^n_{i\to j}\right)^{-1} E^n_{i\to j}\right) \boldsymbol{w}^n_i}.
\end{aligned}
$$

Optimization of the solution

Then the difference vector can be written as

$$\boldsymbol{d}_{j\Gamma}^{n+1} = \frac{(\boldsymbol{w}_i^n)^\top \left(\boldsymbol{u}_{j\Gamma}^n - \boldsymbol{u}_{i\Gamma}^{n-1}\right)}{(\boldsymbol{w}_i^n)^\top \left(I - \left(\hat{A} + E_{i\to j}^n\right)^{-1} E_{i\to j}^n\right) \boldsymbol{w}_i^n} \left(\hat{A} + E_{i\to j}^n\right)^{-1} E_{i\to j}^n \boldsymbol{w}_i^n$$

$$= \left(\hat{A} + E_{i\to j}^n\right)^{-1} E_{i\to j}^n \boldsymbol{w}_i^n \gamma.$$

Meanwhile, the residual of the equation is $\boldsymbol{u}_{j\Gamma}^n - \boldsymbol{u}_{i\Gamma}^{n-1}$. The Galerkin condition is satisfied by $W_i^n \boldsymbol{y}$, where $\boldsymbol{y}$ solves

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

Since $E_{i\to j}^n$ has nullspace W_i^{n-1}, the matrix of this system is upper triangular and we can show the last diagonal entry is $(\boldsymbol{w}_i^n)^\top (I - \left(\hat{A} + E_{i\to j}^n\right)^{-1} E_{i\to j}^n) \boldsymbol{w}_i^n$.

Optimization of the solution

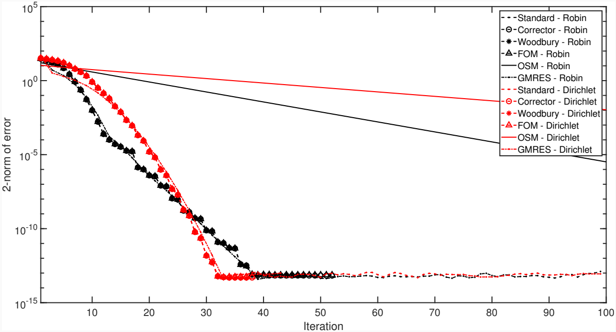
Thus, $\boldsymbol{x} = \gamma \boldsymbol{w}_i^n + W_i^{n-1}\tilde{\boldsymbol{y}}$, where $\tilde{\boldsymbol{y}}$ is $\boldsymbol{y}$ without its last entry. When applying $\left(\hat{A} + E_{i\to j}^n\right)^{-1} E_{i\to j}^n$ to $\boldsymbol{x}$, the nullspace of $E_{i\to j}^n$ eliminates the contribution of W_i^{n-1}, leaving only

$$\left(\hat{A} + E_{i\to j}^n\right)^{-1} E_{i\to j}^n \boldsymbol{x} = \left(\hat{A} + E_{i\to j}^n\right)^{-1} E_{i\to j}^n \boldsymbol{w}_i^n \gamma = \boldsymbol{d}_{j\Gamma}^{n+1}.$$

Thus, the difference vectors are chosen to satisfy an implicit Galerkin condition. $\square$

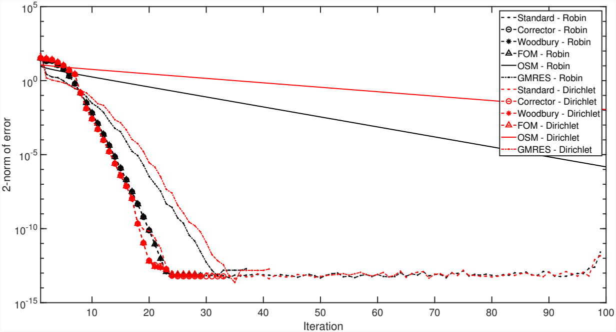
Numerical results

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



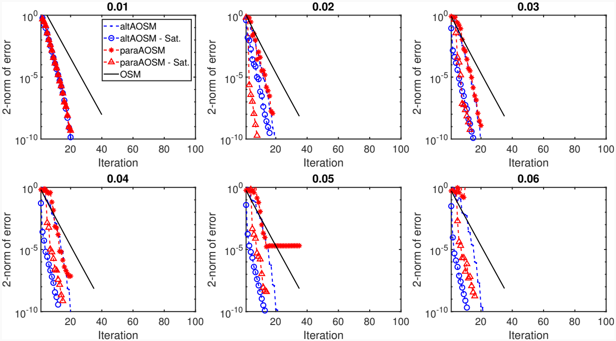
Numerical results

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



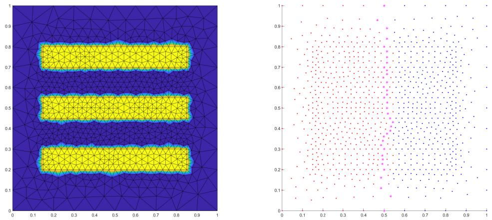
Numerical results

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



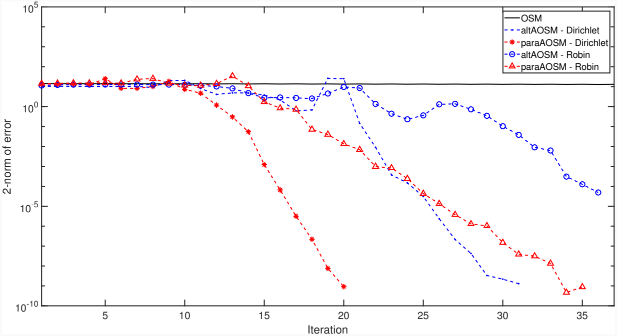
Numerical results

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



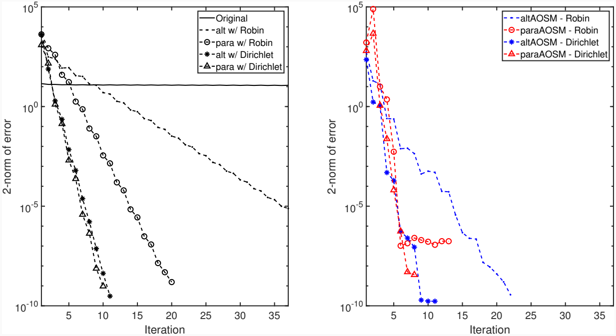
Numerical results

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



Numerical results

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



Conclusions and Future Work

- AOSMs give convergence rates comparable to Krylov subspace methods, but without the extra computations.
- Transmission conditions from AOSMs can be re-used to give fast convergence when repeated solves are required.
- The paraAOSM has some stability issues that need to be fixed.
- Nested decompositions will allow more subdomains, but crosspoints need to be resolved for more general decompositions.