

# Adaptive optimized Schwarz methods

---

Conor McCoid

January 24th, 2025

McMaster University

## **Brief history of domain decomposition**

---

## **Alternate minimization in phase-field fracture models**

---

# **Adaptive optimized Schwarz methods**

---

# Schwarz method with adaptive transmission conditions

Let  $T_{i \rightarrow j}$  change at each iteration, so that the transmission conditions adapt. We can formulate such a Schwarz method, acting only on the difference in successive solutions, as

$$\begin{bmatrix} A_{ii} & A_{i\Gamma} \\ A_{\Gamma i} & A_{\Gamma\Gamma} + T_{j \rightarrow i}^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{d}_i^{n+1} \\ \mathbf{d}_{i\Gamma}^{n+1} \end{bmatrix} = \begin{bmatrix} -A_{\Gamma j} \mathbf{d}_j^n + T_{j \rightarrow i}^{n+1} \mathbf{d}_{j\Gamma}^n - \Delta T_{j \rightarrow i}^n (\mathbf{u}_{i\Gamma}^n - \mathbf{u}_{j\Gamma}^n) \end{bmatrix}$$

where  $i = 1, 2$ ,  $j = 3 - i$ , and  $\Delta T_{j \rightarrow i}^n$  represents the update to the transmission condition  $T_{j \rightarrow i}^n$  at this step,

$$T_{j \rightarrow i}^{n+1} = T_{j \rightarrow i}^n + \Delta T_{j \rightarrow i}^n.$$

## Condensing the iteration

We can condense these iterations to express them as only acting on the difference at the interface. First we note from the first row of blocks that

$$\mathbf{d}_i^{n+1} = -A_{ii}^{-1} A_{i\Gamma} \mathbf{d}_{i\Gamma}^{n+1},$$

and so likewise

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

Combining with the second row of blocks gives

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

## Difference between $T$ and $S$

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

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

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

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

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

## Action of $E$

The matrix-vector product  $E_{i \rightarrow j}^{n+1} \mathbf{d}_{i\Gamma}^n$ , equal to

$$E_{i \rightarrow j}^{n+1} \mathbf{d}_{i\Gamma}^n = -A_{\Gamma i} \mathbf{d}_i^n + T_{i \rightarrow j}^{n+1} \mathbf{d}_{i\Gamma}^n,$$

is computed in the course of the Schwarz method. Thus, we will have a sequence of vector pairs,  $(\mathbf{d}_{i\Gamma}^n, E_{i \rightarrow j}^{n+1} \mathbf{d}_{i\Gamma}^n)$  which we can use to approximate  $E$  without ever calculating it.

Each vector pair gives a rank one approximation of  $E$ :

$$E_{i \rightarrow j}^{n+1} \approx \frac{E_{i \rightarrow j}^{n+1} \mathbf{d}_{i\Gamma}^n}{\|\mathbf{d}_{i\Gamma}^n\|_2^2}.$$

To combine these rank one matrices, we apply **modified Gram-Schmidt** to the vectors  $\mathbf{d}$  and a commensurate process to the vectors  $E\mathbf{d}$ .



---

**Algorithm 1** Iterative action approximation

---

 $[V_n, W_n] = \text{IAA}(\{\mathbf{x}_k\}_{k=1}^n, E)$ 

---

- 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:      $\mathbf{w}_k := \mathbf{x}_k$ ,  $\mathbf{v}_k := E \mathbf{x}_k$
  - 5:     **for**  $i = 1 : k - 1$  **do**
  - 6:          $h \leftarrow \langle \mathbf{w}_i, \mathbf{w}_k \rangle$ ,  $\mathbf{w}_k \leftarrow \mathbf{w}_k - h \mathbf{w}_i$
  - 7:          $\mathbf{v}_k \leftarrow \mathbf{v}_k - h \mathbf{v}_i$
  - 8:     **end for**
  - 9:      $\alpha_k := 1 / \|\mathbf{w}_k\|$ ,  $\mathbf{w}_k \leftarrow \alpha_k \mathbf{w}_k$ ,  $\mathbf{v}_k \leftarrow \alpha_k \mathbf{v}_k$
  - 10: **end for**
  - 11: Outputs:  $W_n = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_n \end{bmatrix}$ ,  $V_n = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \end{bmatrix}$
  - 12:  $V_n W_n^\top \approx E$
-

## Lemma

*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  $[V_n, W_n] = \text{IAA}(\mathcal{S}, E)$  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*

$$E^{k+1} = E(I - W_k W_k^\top),$$

$$\mathbf{v}_k = E \mathbf{w}_k = E^k \mathbf{w}_k$$

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

## Choice of adaptive transmission conditions

For our adaptive transmission conditions, we want

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

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

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

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

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

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

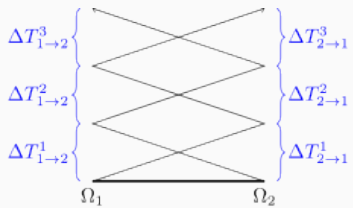
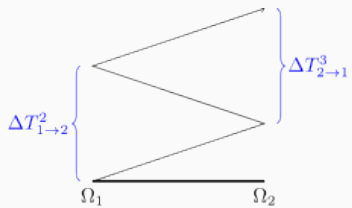
at every step.

---

**Algorithm 2** altAOSM: AOSM applied to multiplicative Schwarz

---

- 1: Start with initial transmission conditions  $T_{1 \rightarrow 2}^1$  and  $T_{2 \rightarrow 1}^1$
  - 2: Make initial guess  $\mathbf{u}_{1\Gamma}^0$
  - 3: Calculate  $\mathbf{u}_1^0 = A_{11}^{-1}(\mathbf{f}_1 - A_{1\Gamma}\mathbf{u}_{1\Gamma}^0)$
  - 4: Solve for  $\mathbf{u}_2^1$  and  $\mathbf{u}_{2\Gamma}^1$ , then for  $\mathbf{u}_1^2$  and  $\mathbf{u}_{1\Gamma}^2$
  - 5: Calculate  $\mathbf{d}_{1\Gamma}^2 = \mathbf{u}_{1\Gamma}^2 - \mathbf{u}_{1\Gamma}^0$  and  $\mathbf{d}_1^2$  and set  $n = 2$
  - 6: **while**  $\|\mathbf{d}_{1\Gamma}^n\| + \|\mathbf{d}_{2\Gamma}^{n-1}\| \geq tol$  **do**
  - 7:     **for**  $i = 1 : 2$  **do**
  - 8:         Run an iteration of IAA
  - 9:         Set  $\Delta T_{i \rightarrow j}^n = -\mathbf{v}_i^n(\mathbf{w}_i^n)^\top$
  - 10:         Solve for  $\mathbf{d}_{j\Gamma}^{n+1}$  and  $\mathbf{d}_j^{n+1}$
  - 11:          $\mathbf{u}_j^{n+1} := \mathbf{u}_j^{n-1} + \mathbf{d}_j^{n+1}$ ,  $\mathbf{u}_{j\Gamma}^{n+1} := \mathbf{u}_{j\Gamma}^{n-1} + \mathbf{d}_{j\Gamma}^{n+1}$
  - 12:          $n \leftarrow n + 1$
  - 13:     **end for**
  - 14: **end while**
  - 15: Output:  $\mathbf{u} = [\mathbf{u}_1^n ; (\mathbf{u}_{1\Gamma}^n + \mathbf{u}_{2\Gamma}^{n-1})/2 ; \mathbf{u}_2^{n-1}]$
-



## Theorem

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

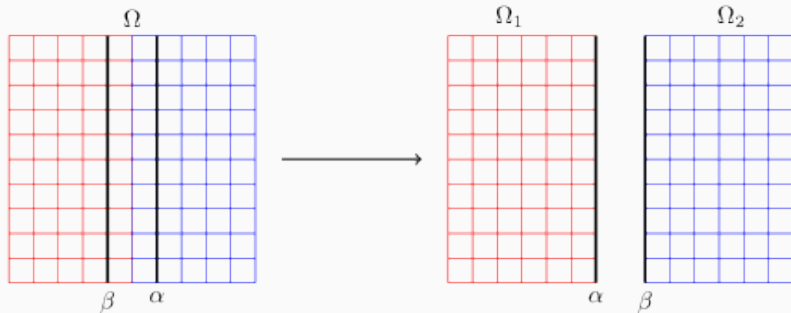
$$\mathbf{d}_{j\Gamma}^{n+1} = \left( \hat{A} + E_{i \rightarrow j}^n \right)^{-1} E_{i \rightarrow j}^n \mathbf{x},$$

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

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

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

# Algebraic Schwarz



Consider the following algebraic system with block tridiagonal matrix  $A$ :

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

The solution is divided into two subdomains:  $\Omega_1$  contains the variables in  $\mathbf{u}_1$  and  $\mathbf{u}_\Gamma$ , and  $\Omega_2$  contains the variables in  $\mathbf{u}_2$  and  $\mathbf{u}_\Gamma$ . The variables  $\mathbf{u}_\Gamma$  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

$$\begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T_{2 \rightarrow 1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{n+1} \\ \mathbf{u}_{1\Gamma}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \end{bmatrix} + \begin{bmatrix} -A_{\Gamma 2} \mathbf{u}_2^n + T_{2 \rightarrow 1} \mathbf{u}_{2\Gamma}^n \\ 0 \end{bmatrix},$$
$$\begin{bmatrix} A_{22} & A_{2\Gamma} \\ A_{\Gamma 2} & A_{\Gamma\Gamma} + T_{1 \rightarrow 2} \end{bmatrix} \begin{bmatrix} \mathbf{u}_2^{n+1} \\ \mathbf{u}_{2\Gamma}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_\Gamma \end{bmatrix} + \begin{bmatrix} 0 \\ -A_{\Gamma 1} \mathbf{u}_1^n + T_{1 \rightarrow 2} \mathbf{u}_{1\Gamma}^n \end{bmatrix},$$

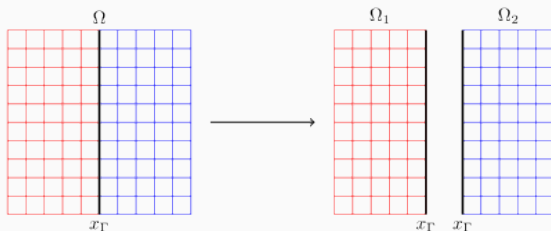
for any choice of matrices  $T_{1 \rightarrow 2}$  and  $T_{2 \rightarrow 1}$  (that does not make these systems singular).

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  $T$ .



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:

$$\frac{\partial u_1^n}{\partial x} - pu_1^n(0, y) = \frac{\partial u_2^{n-1}}{\partial x} - pu_2^{n-1}(0, y),$$

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:

$$\frac{\partial u_1^n}{\partial x} - pu_1^n(0, y) + q\frac{\partial u_1^n}{\partial y} = \frac{\partial u_2^{n-1}}{\partial x} - pu_2^{n-1}(0, y) + q\frac{\partial u_2^{n-1}}{\partial y},$$

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:

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

With these, algebraic Schwarz with two subdomains converges in

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:

$$\begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} + T_{2 \rightarrow 1}^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{n+1} \\ \mathbf{u}_{1\Gamma}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \end{bmatrix} - \begin{bmatrix} A_{\Gamma 2} \mathbf{u}_2^n \\ A_{\Gamma 1} \mathbf{u}_1^n \end{bmatrix} + \begin{bmatrix} T_{2 \rightarrow 1}^{n+1} \mathbf{u}_{2\Gamma}^n \\ T_{1 \rightarrow 2}^{n+1} \mathbf{u}_{1\Gamma}^n \end{bmatrix},$$
$$\begin{bmatrix} A_{22} & A_{2\Gamma} \\ A_{\Gamma 2} & A_{\Gamma\Gamma} + T_{1 \rightarrow 2}^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_2^{n+1} \\ \mathbf{u}_{2\Gamma}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_\Gamma \end{bmatrix} - \begin{bmatrix} A_{\Gamma 1} \mathbf{u}_1^n \\ A_{\Gamma 2} \mathbf{u}_2^n \end{bmatrix} + \begin{bmatrix} T_{2 \rightarrow 1}^{n+1} \mathbf{u}_{2\Gamma}^n \\ T_{1 \rightarrow 2}^{n+1} \mathbf{u}_{1\Gamma}^n \end{bmatrix}.$$

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

## 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:

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



## Adaptive Transmission Conditions

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

$$E_{i \rightarrow j}^{n+1} \mathbf{d}_{i\Gamma}^n = -A_{\Gamma i} \mathbf{d}_i^n + T_{i \rightarrow j}^{n+1} \mathbf{d}_{i\Gamma}^n.$$

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

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

## Adaptive Transmission Conditions

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

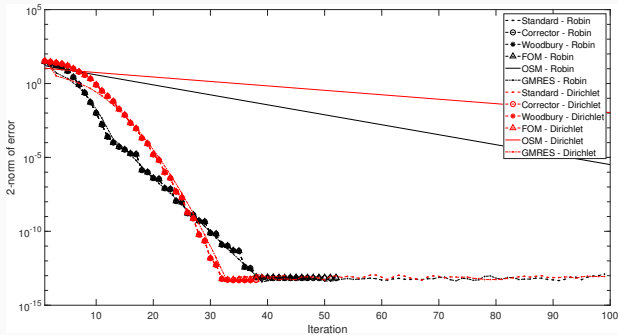
$$E \approx \frac{\mathbf{y}\mathbf{d}^\top}{\|\mathbf{d}\|_2^2}.$$

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

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

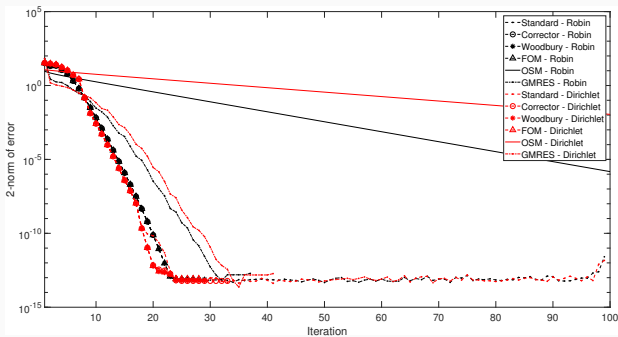
# Numerical results

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



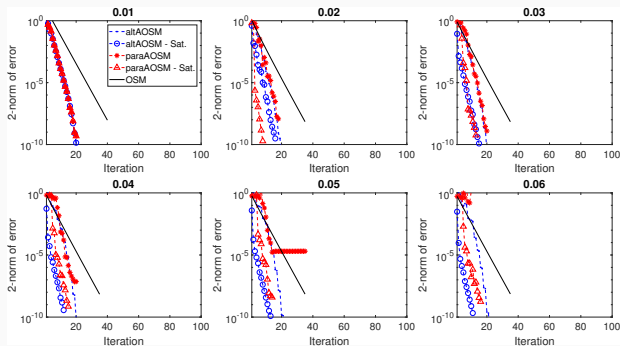
# Numerical results

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



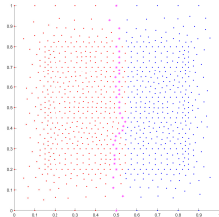
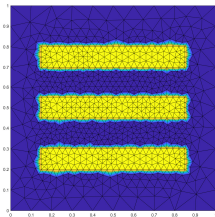
# Numerical results

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



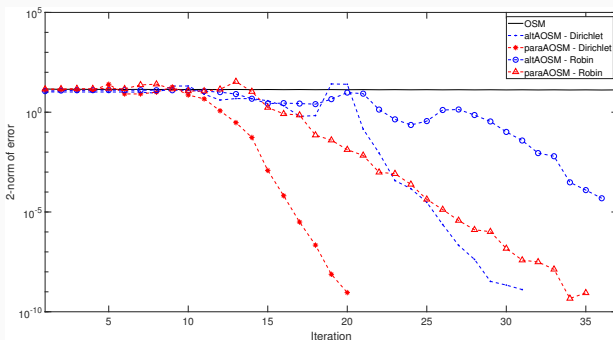
# Numerical results

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



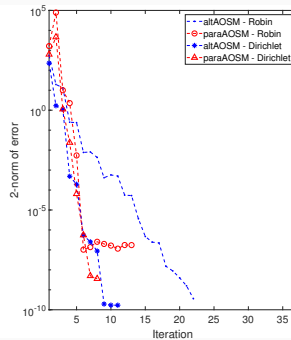
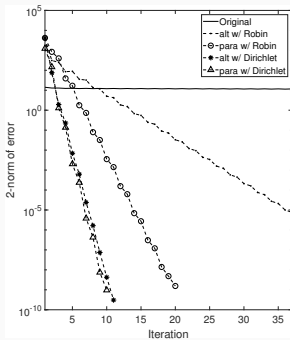
# Numerical results

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



# Numerical results

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





- 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.