Adaptive optimized Schwarz methods

Conor McCoid

January 24th, 2025

McMaster University
Joint work with Felix Kwok at the Université Laval

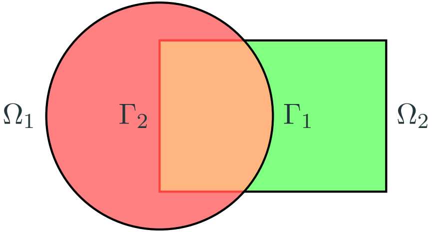
Outline

Introduction to domain decomposition

H.A. Schwarz, 1869

How do we solve the Laplace equation on complicated domains?

We split the domain into simpler subdomains.

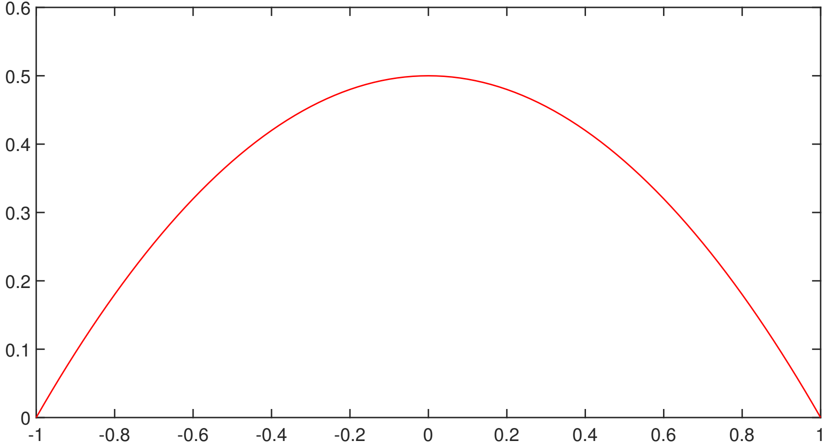


Alternating Schwarz method:

$$\begin{cases} \Delta u_1^{n+1} = 0 & \text{in } \Omega_1, \\ u_1^{n+1} = u_2^n & \text{on } \Gamma_1, \end{cases} \qquad \begin{cases} \Delta u_2^{n+1} = 0 & \text{in } \Omega_2, \\ u_2^{n+1} = u_1^{n+1} & \text{on } \Gamma_2. \end{cases}$$

Simple example

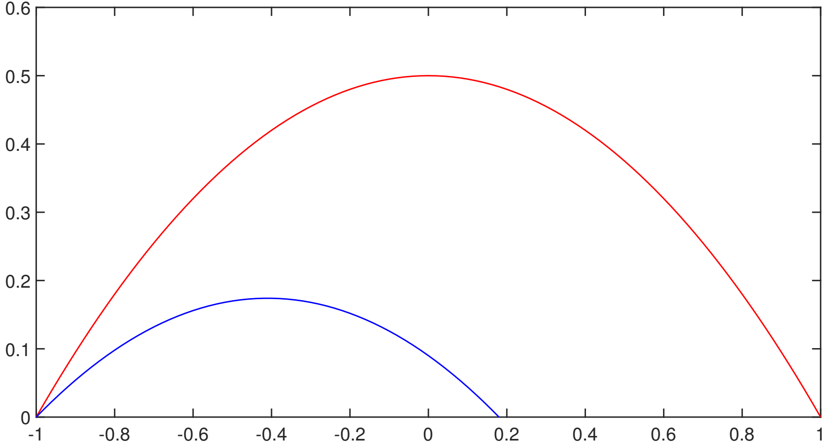
$$u''(x) = -1, \quad x \in [-1, 1], \quad u(-1) = u(1) = 0$$
$$\Omega_1 = [-1, 0.18], \quad \Omega_2 = [-0.22, 1]$$



Simple example

$$u''(x) = -1, \quad x \in [-1, 1], \quad u(-1) = u(1) = 0$$

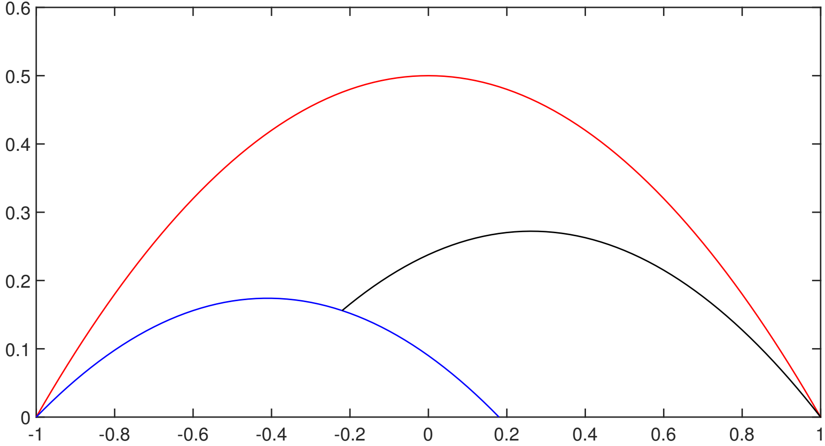
$$\Omega_1 = [-1, 0.18], \quad \Omega_2 = [-0.22, 1]$$



Simple example

$$u''(x) = -1, \quad x \in [-1, 1], \quad u(-1) = u(1) = 0$$

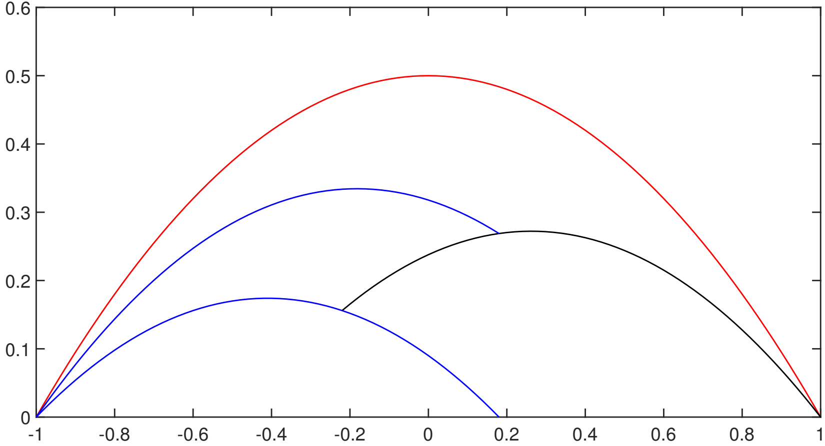
$$\Omega_1 = [-1, 0.18], \quad \Omega_2 = [-0.22, 1]$$



Simple example

$$u''(x) = -1, \quad x \in [-1, 1], \quad u(-1) = u(1) = 0$$

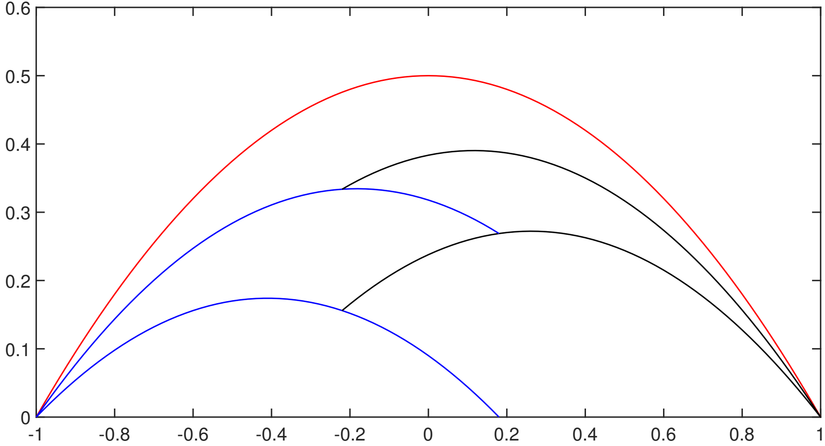
$$\Omega_1 = [-1, 0.18], \quad \Omega_2 = [-0.22, 1]$$



Simple example

$$u''(x) = -1, \quad x \in [-1,1], \quad u(-1) = u(1) = 0$$

$$\Omega_1 = [-1, 0.18], \quad \Omega_2 = [-0.22, 1]$$



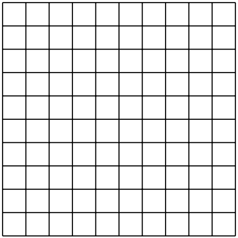
Discretization of continuous problem

Suppose we have an elliptic PDE:

$$\begin{cases} \Delta u(x,y) = f(x,y), & x, y \in \Omega, \\ u(x,y) = g(x,y), & x, y \in \partial\Omega, \end{cases}$$

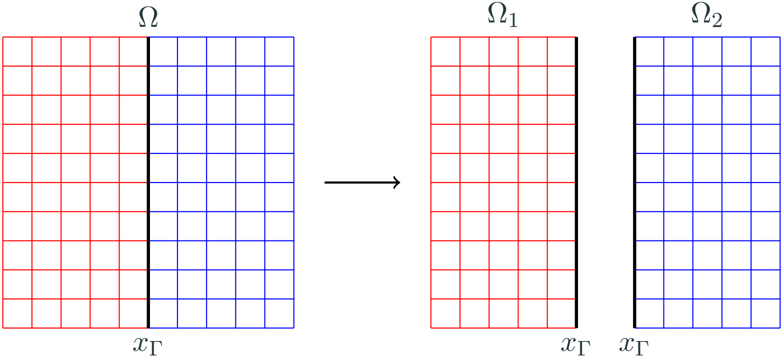
that we solve with finite differences on a structured grid:

$$\Delta u_{i,j} \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2}.$$



Domain decomposition

We split the domain into two along a line $x = x_\Gamma$. This gives us two domains Ω_1 and Ω_2, as well as the interface between them, Γ.



Discrete problem

The discrete problem can be represented by a block tridiagonal system:

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

where the variables $\boldsymbol{u}_1$ represent the solution in the interior of Ω_1, $\boldsymbol{u}_2$ those in Ω_2, and $\boldsymbol{u}_\Gamma$ those on the interface.

Algebraic decomposition

The subproblem on Ω_1 is

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

and the subproblem on Ω_2 is

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

$\boldsymbol{u}_\Gamma$ appears in both subproblems since the interface is shared between the two subdomains. These copies are distinct and must be recombined in some way at the end, for example:

$$\boldsymbol{u}_\Gamma = \frac{\boldsymbol{u}_{1\Gamma} + \boldsymbol{u}_{2\Gamma}}{2}.$$

Transmission conditions

Transmission conditions are boundary conditions of the subdomains. Algebraically, they are implemented through the matrices $T_{1\to 2}$ (which dictates how information passes *from* Ω_1 *to* Ω_2) and $T_{2\to 1}$ (Ω_2 to Ω_1).

So far, we've seen Dirichlet boundary conditions on the subdomains, which is equivalent to $T = 0$.

Options for transmission conditions

- Dirichlet boundary conditions, equivalent to $T = 0$
- Neumann boundary conditions, allow minimal overlap
- Optimized Robin boundary conditions:

$$\frac{\partial \boldsymbol{u}_1^{n+1}}{\partial x} - p\boldsymbol{u}_1^{n+1} = \frac{\partial \boldsymbol{u}_2^{n}}{\partial x} - p\boldsymbol{u}_2^{n}$$

  for some p optimized using Fourier analysis
- Absorbing boundary conditions, equivalent to using Schur complements:

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

With absorbing boundary conditions, the method becomes direct. However, the Schur complements are expensive to compute and in general dense.

Overview

Schwarz methods play an important role in parallel computing.

The current research directions for domain decomposition methods include issues with

- crosspoints (where three or more subdomains meet);
- scalability (adding more processors should improve efficiency), and;
- convergence rates (effectiveness of each iteration).

The work presented here aims to tackle this last point by finding new types of transmission conditions.

Adaptive optimized Schwarz methods

Schwarz method with adaptive transmission conditions

Let $T_{i\to j}$ change at each iteration, so that the transmission conditions adapt. We can formulate such a Schwarz method, acting only on $\boldsymbol{d}_i^{n+1} = \boldsymbol{u}_i^{n+1} - \boldsymbol{u}_i^n$, as

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

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

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

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

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

and so likewise

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

Combining with the second row of blocks gives

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

(Recall $S_{i\to j} = -A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma}$)

Difference between T and S

Notice the difference between the T and S matrices 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 $\hat{A} := A_{\Gamma\Gamma} + S_{1\to 2} + S_{2\to 1}$, the Schur complement of the full system, 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.

Action of E

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

A standard Schwarz method will compute vector pairs $(\boldsymbol{d}, E\boldsymbol{d})$.

The vectors $\boldsymbol{d}$ are needed to find the solutions, while the vectors $E\boldsymbol{d}$ are found in the right hand sides of the systems to be solved. We can use these vector pairs to approximate E without calculating it.

Each vector pair gives a rank one approximation of E:

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

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

Algorithm 1 Iterative action approximation (IAA)

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

Inputs: $\{\boldsymbol{d}_k\}_{k=1}^{n} \subset \mathbb{R}^M$, $E \in \mathbb{R}^{M\times M}$
$\alpha_1 := 1/\left\|\boldsymbol{d}_1\right\|$, $\boldsymbol{w}_1 := \alpha_1 \boldsymbol{d}_1$, $\boldsymbol{v}_1 := \alpha_1 E \boldsymbol{d}_1$
for $k = 2 : n$ **do**
    $\boldsymbol{w}_k := \boldsymbol{d}_k$, $\boldsymbol{v}_k := E\boldsymbol{d}_k$
    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
$W_n = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_n \end{bmatrix}$, $V_n = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_n \end{bmatrix}$
$V_n W_n^\top \approx E$

Lemma

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.

That is, the updates to E increase its nullspace and preserve the relation between $\boldsymbol{w}$ and $\boldsymbol{v}$, which is the relation between $\boldsymbol{d}$ and $E\boldsymbol{d}$.

Other fun facts about the IAA

- The vectors $\boldsymbol{w}_k$ form an orthonormal basis of a Krylov subspace
- Each subdomain has its own set of $\boldsymbol{w}_k$ and its own Krylov subspace
- The vectors $\boldsymbol{v}_k$ are *not* orthogonal

Choice of adaptive transmission conditions

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

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

at every step.

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 $\boldsymbol{u}^1_{2\Gamma}$, then 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 $i = 1:2$ (and $j = 3 - i$) **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 $\boldsymbol{d}^{n+1}_j$

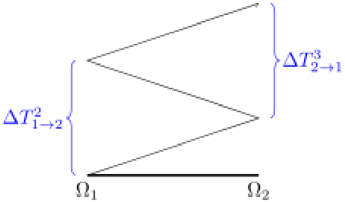
    $\boldsymbol{u}^{n+1}_j := \boldsymbol{u}^{n-1}_j + \boldsymbol{d}^{n+1}_j$, $\boldsymbol{u}^{n+1}_{j\Gamma} := \boldsymbol{u}^{n-1}_{j\Gamma} + \boldsymbol{d}^{n+1}_{j\Gamma}$

    $n \leftarrow n + 1$

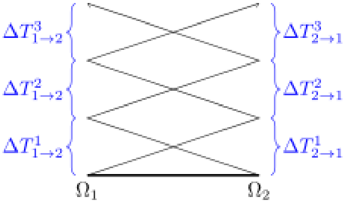
  end for

end while

Output: $\boldsymbol{u} = [\boldsymbol{u}^n_1\ ;\ (\boldsymbol{u}^n_{1\Gamma} + \boldsymbol{u}^{n-1}_{2\Gamma})/2\ ;\ \boldsymbol{u}^{n-1}_2]$



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

AOSM 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 $\operatorname{span}(W_i^n)$.

AOSM Galerkin condition

1. The solution on the interface, $\boldsymbol{u}_\Gamma$, solves a system $B\boldsymbol{u}_\Gamma = \boldsymbol{b}$
2. There is a vector $\boldsymbol{x} \in \operatorname{span}(W_i^n)$ such that $B(\boldsymbol{u}_{i\Gamma}^{n-1} + \boldsymbol{x}) - \boldsymbol{b}$ is orthogonal to $\operatorname{span}(W_i^n)$
3. The vector $\boldsymbol{d}_{j\Gamma}^{n+1}$, determined by the AOSM, is equal to

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

Numerical results of AOSM

Simple comparison with other methods

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

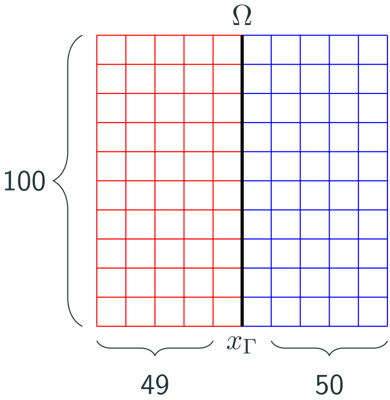


Figure 1: 100×100 evenly spaced grid split into two subdomains along the 50th value of x

Simple comparison with other methods

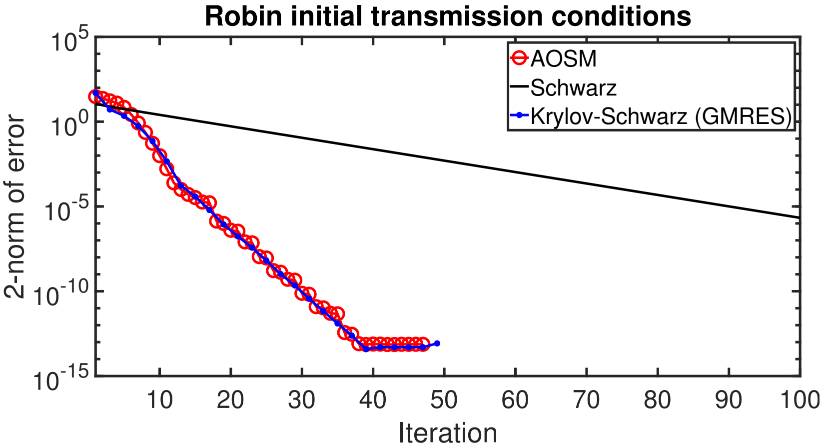


Figure 2: Comparison between Schwarz, AOSM and Krylov-Schwarz on simple elliptic PDE

Precursor to multiple subdomains: red-black decompositions

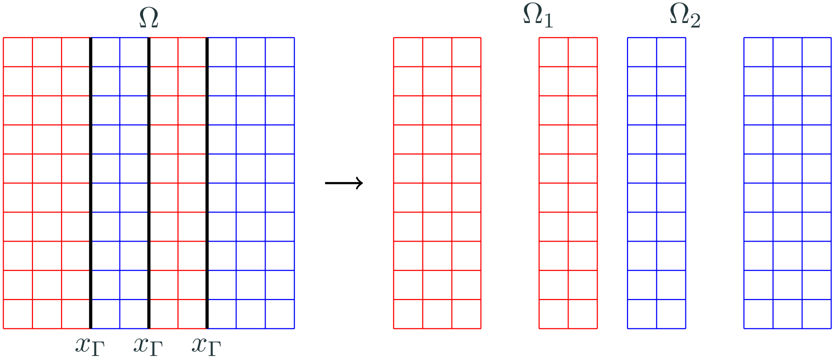


Figure 3: Splitting the 100×100 grid into four strips, then pairing the strips into two algebraic subdomains

Comparison: stripwise

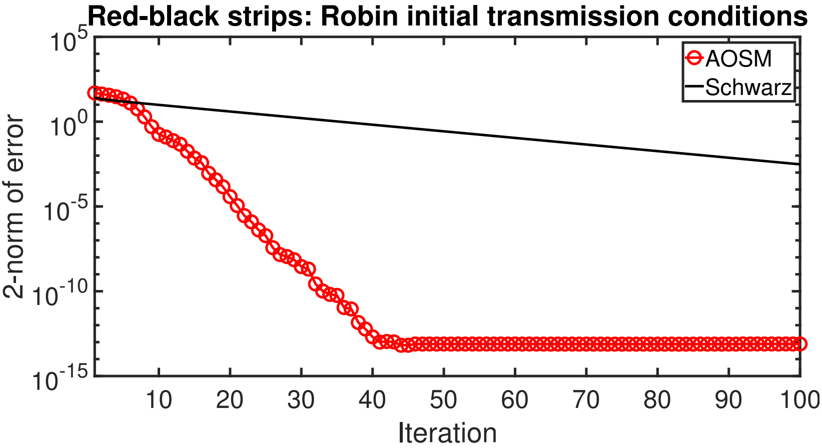


Figure 4: Convergence for AOSM and Schwarz on the stripwise decomposition

Adapted transmission conditions: stripwise

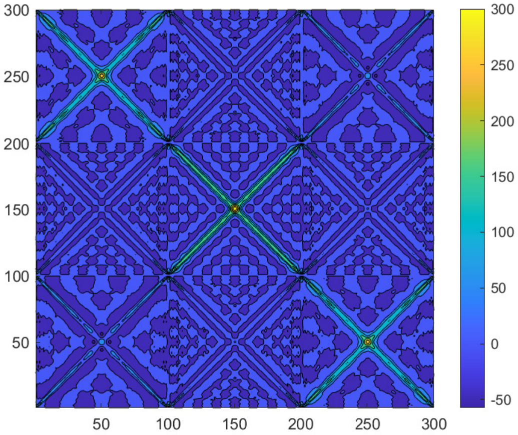


Figure 5: Matrix $T_{1\to 2}$ from AOSM after convergence of stripwise example

Heterogeneous elliptic PDE

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

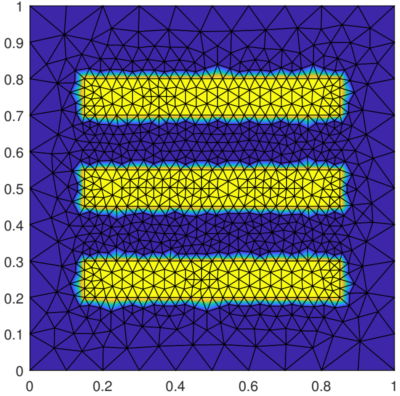


Figure 6: $\alpha(x,y)=1$ except along three thin channels where $\alpha(x,y)=1000$

Unstructured grid

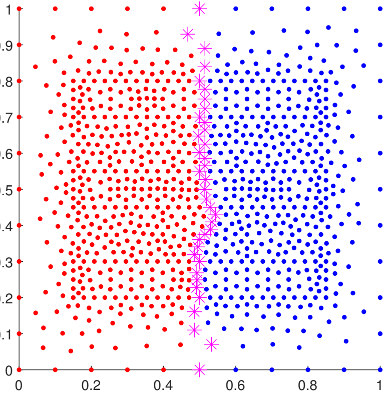


Figure 7: Splitting an unstructured grid into two subdomains

First round of solves

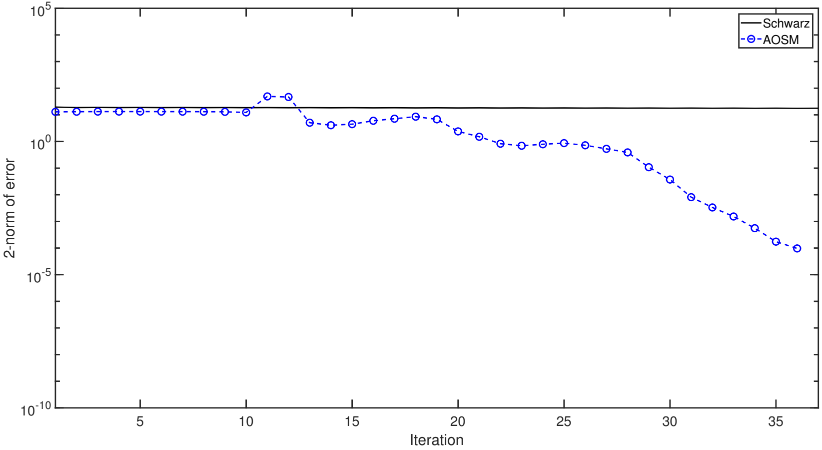


Figure 8: (Lack of) convergence for Schwarz and AOSM for heterogeneous elliptic PDE

Second round of solves

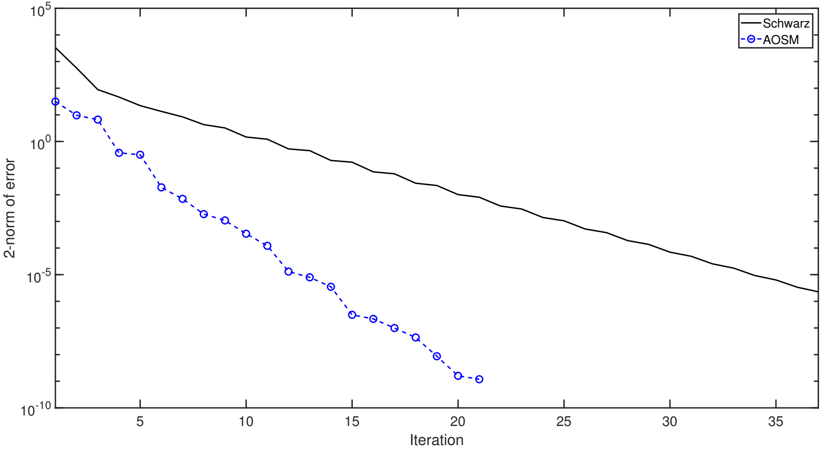


Figure 9: Convergence for Schwarz and AOSM using adapted transmission conditions

Future work

AOSM difficulties: Crosspoints

We need to generalize the AOSM for multiple subdomains. We've already seen that we need new types of updates to the transmission conditions to account for the shape of the Schur complements.

A common problem with domain decomposition methods for multiple subdomains is **crosspoints**: points where three or more subdomains meet. At this point, it is not clear how information should be transmitted.

Some new results have figured out workarounds for rectilinear grids (Chaudet-Dumas & Gander, 2023). More general techniques are needed.

AOSM difficulties: Scalability

Another common issue with domain decomposition methods is **scalability**. A method in parallel computing is scalable if adding more processors always improves efficiency.

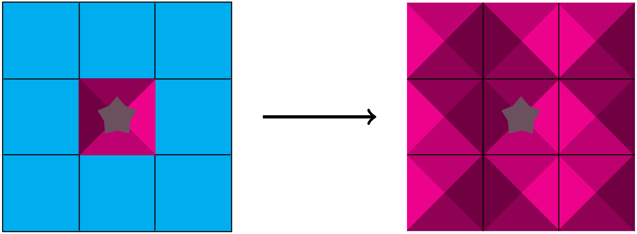
For domain decomposition methods, information from each subdomain must pass to every other subdomain, often multiple times. This means that the computation time is limited by the length of the longest path through the network of subdomains.

Researchers have found multigrid methods can be combined with domain decomposition methods to overcome much of this issue.

Symmetrized cells

As an alternative way to use AOSMs for multiple subdomains, one can first prioritize optimizing the transmission conditions for each subproblem.

Isolate the subdomain and symmetrize it, i.e. make one or several *virtual* copies of the subdomain. Solve this subproblem repeatedly until the T matrices have adapted. Then redistribute the T matrices to their appropriate subproblems.



Conclusions

- AOSMs give Krylov-Schwarz convergence rates without the extra computations.
- Transmission conditions from AOSMs can be re-used to give fast convergence.
- We need to generalize the AOSMs for different types of approximations of the Schur complement.
- Symmetrized cells can be used to find the transmission conditions beforehand.