Adaptive optimized Schwarz methods

Conor McCoid

January 24th, 2025

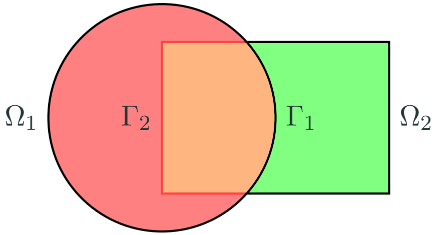
McMaster University

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.



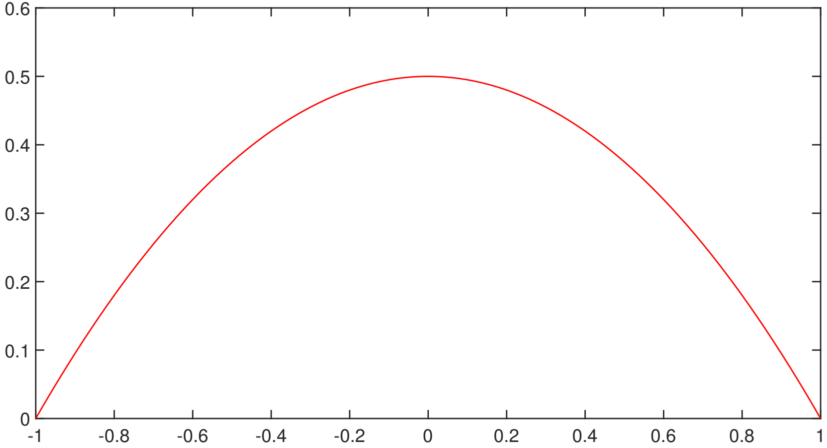
Alternating Schwarz method:

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

Simple example

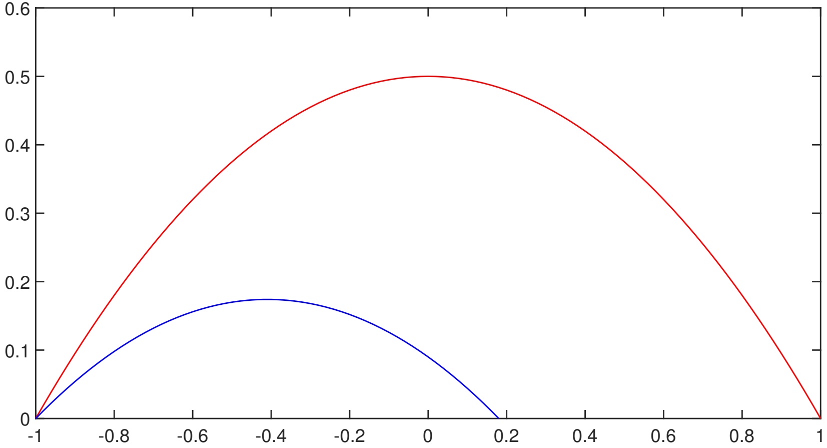
$$u''(x) = -1, \quad x \in [-1, 1], \quad u(-1) = u(1) = 0$$

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



Simple example

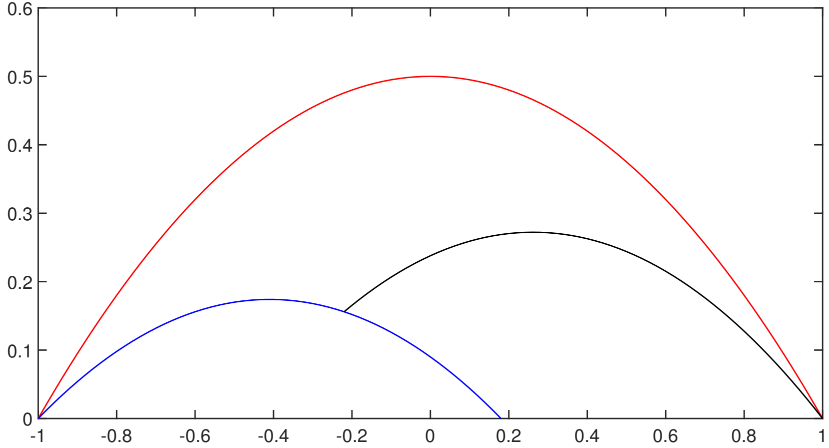
$$u''(x) = -1, \quad x \in [-1, 1], \quad u(-1) = u(1) = 0$$
$$\Omega_1 = [-1, 0.18], \quad \Omega_2 = [-0.22, 1]$$



Simple example

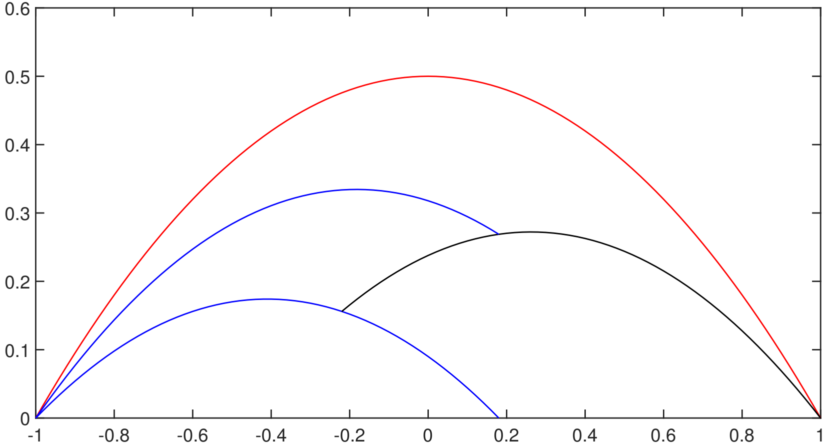
$$u''(x) = -1, \quad x \in [-1, 1], \quad u(-1) = u(1) = 0$$

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



Simple example

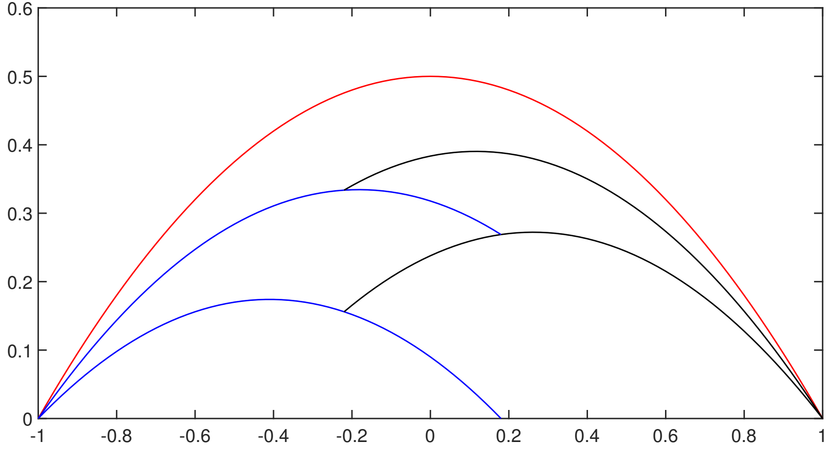
$$u''(x) = -1, \quad x \in [-1, 1], \quad u(-1) = u(1) = 0$$
$$\Omega_1 = [-1, 0.18], \quad \Omega_2 = [-0.22, 1]$$



Simple example

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

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



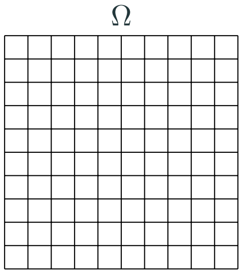
Discretization of continuous problem

Suppose we have an elliptic PDE:

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

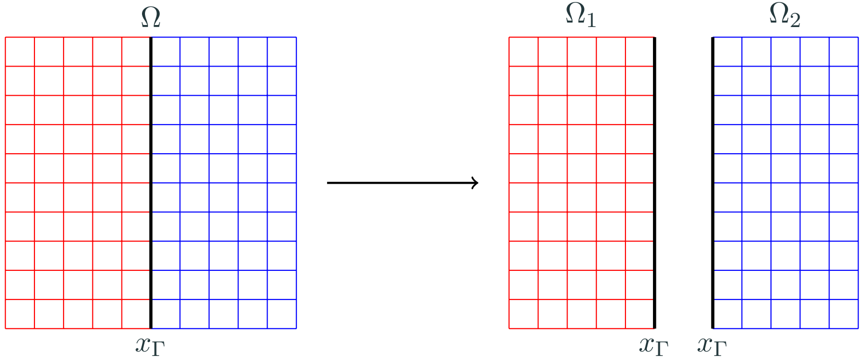
that we solve with finite differences on a structured grid:

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



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 $T_{1\to2}$ (which dictates how information passes *from* Ω_1 *to* Ω_2) and $T_{2\to1}$ (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:

  $$\frac{\partial u_1^{n+1}}{\partial x} - p u_1^{n+1} = \frac{\partial u_2^n}{\partial x} - p u_2^n$$

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

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

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

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 the difference in successive solutions,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

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

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

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

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.

Action of E

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

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

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

Each vector pair gives a rank one approximation of E:

$$E_{i\to j}^{n+1} \approx \frac{E_{i\to j}^{n+1}\boldsymbol{d}_{i\Gamma}^{n}}{\left\|\boldsymbol{d}_{i\Gamma}^{n}\right\|_2^2}.$$

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

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

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

Properties of the IAA

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

Choice of adaptive transmission conditions

For our adaptive transmission conditions, we want

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

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

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:

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

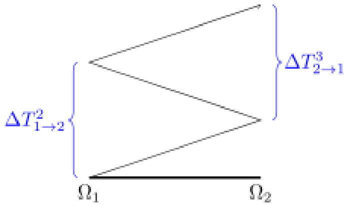
This choice of ΔT eliminates the product $E_{i\to j}^{n+1}\boldsymbol{d}_{i\Gamma}^n$ from the earlier systems. We must now solve

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

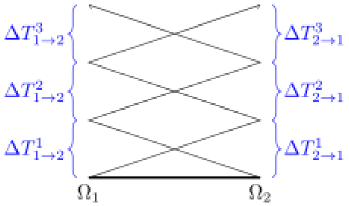
at every step.

Algorithm 2 altAOSM: AOSM applied to multiplicative Schwarz

```
1: Start with initial transmission conditions $T^1_{1\to 2}$ and $T^1_{2\to 1}$
2: Make initial guess $\boldsymbol{u}^0_{1\Gamma}$
3: Calculate $\boldsymbol{u}^0_1 = A^{-1}_{11}(\boldsymbol{f}_1 - A_{1\Gamma}\boldsymbol{u}^0_{1\Gamma})$
4: Solve for $\boldsymbol{u}^1_2$ and $\boldsymbol{u}^1_{2\Gamma}$, then for $\boldsymbol{u}^2_1$ and $\boldsymbol{u}^2_{1\Gamma}$
5: Calculate $\boldsymbol{d}^2_{1\Gamma} = \boldsymbol{u}^2_{1\Gamma} - \boldsymbol{u}^0_{1\Gamma}$ and $\boldsymbol{d}^2_1$ and set $n = 2$
6: while $\|\boldsymbol{d}^n_{1\Gamma}\| + \|\boldsymbol{d}^{n-1}_{2\Gamma}\| \geq tol$ do
7:     for $i = 1:2$ do
8:         Run an iteration of IAA
9:         Set $\Delta T^n_{i\to j} = -\boldsymbol{v}^n_i(\boldsymbol{w}^n_i)^\top$
10:        Solve for $\boldsymbol{d}^{n+1}_{j\Gamma}$ and $\boldsymbol{d}^{n+1}_j$
11:        $\boldsymbol{u}^{n+1}_j := \boldsymbol{u}^{n-1}_j + \boldsymbol{d}^{n+1}_j$, $\boldsymbol{u}^{n+1}_{j\Gamma} := \boldsymbol{u}^{n-1}_{j\Gamma} + \boldsymbol{d}^{n+1}_{j\Gamma}$
12:        $n \leftarrow n + 1$
13:    end for
14: end while
15: 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* $\operatorname{span}(W_i^n)$.

Numerical results

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

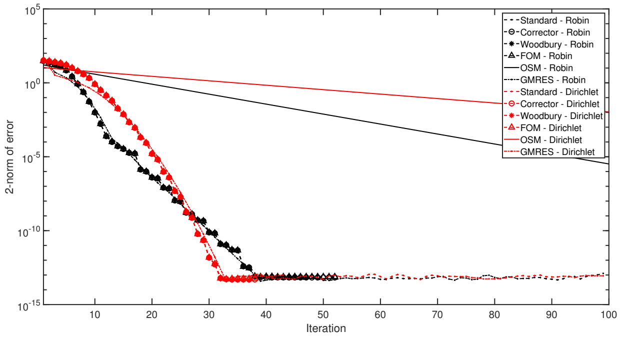
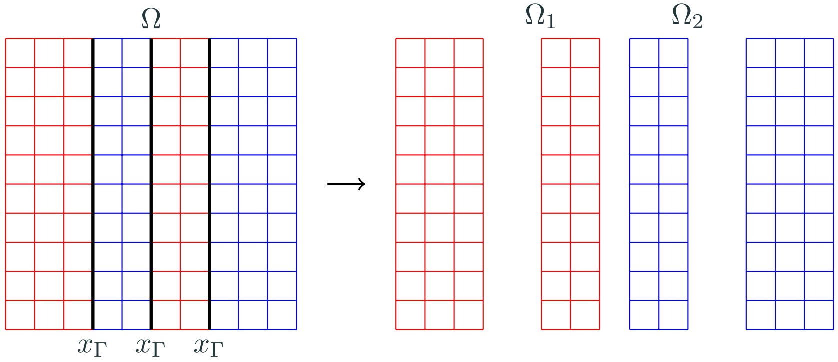
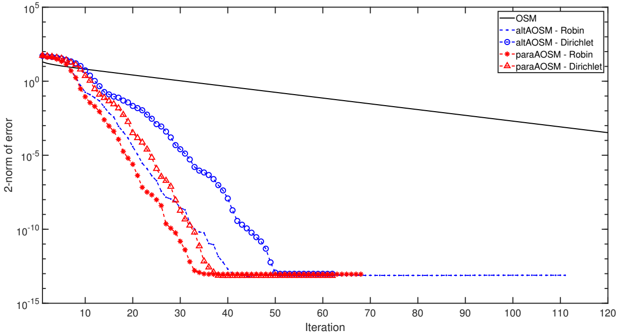


Figure 1: Comparison of altAOSM versions with $N = 10,000$, $M = 100$, using both Robin and Dirichlet boundary conditions.

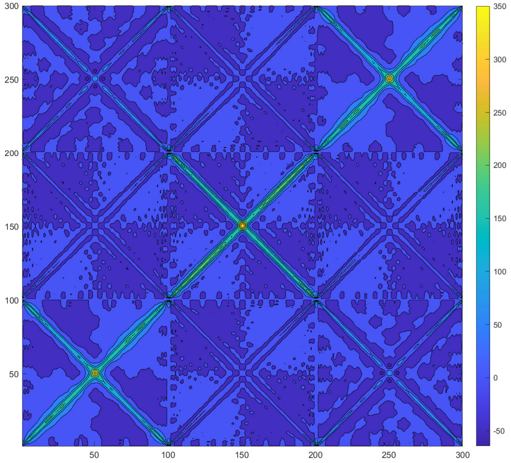
Precursor to multiple subdomains: red-black decompositions



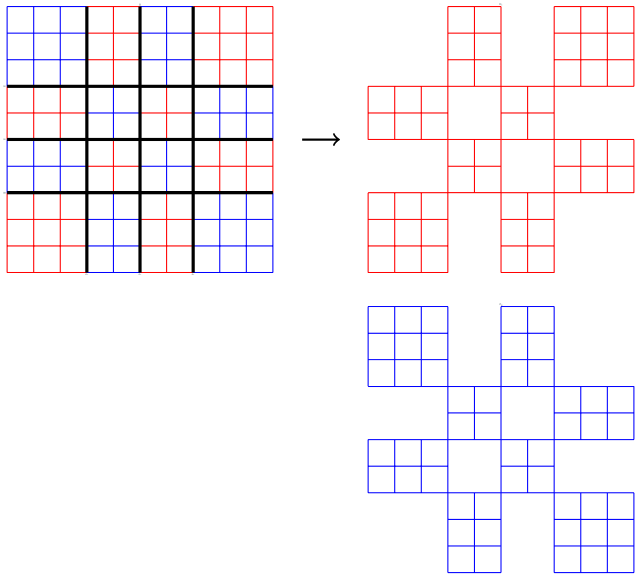
Comparison: stripwise



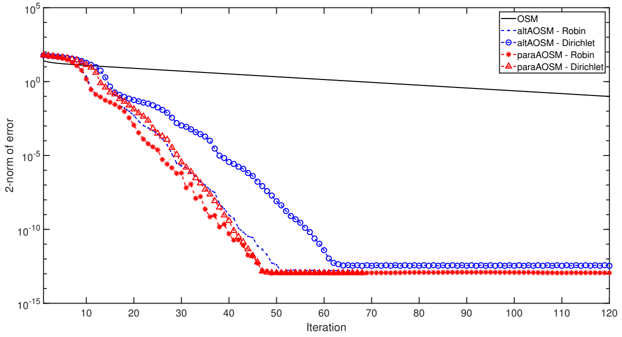
Adapted transmission conditions: stripwise



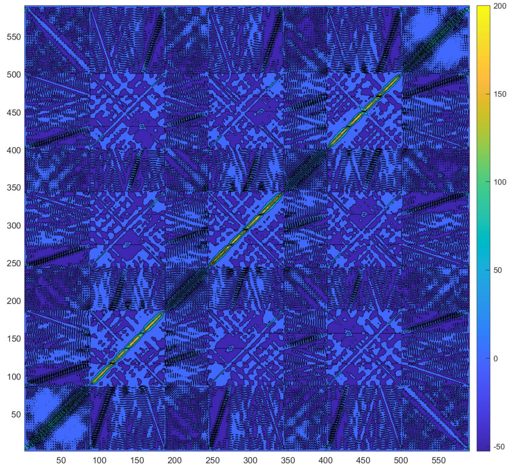
Checkerboard decomposition



Comparison: checkerboard



Adapted transmission conditions: checkerboard



New approximations of Schur complements

It's clear that the transmission conditions we're finding do not need to be so dense. For the multiple subdomains, it appears we can approximate the Schur complements with block diagonal matrices.

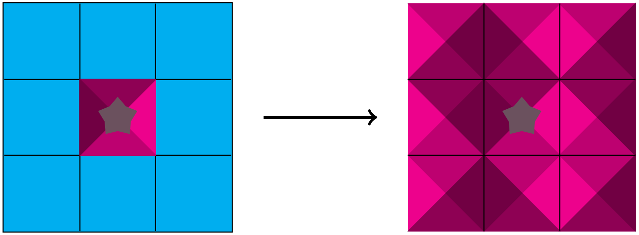
We could also use completely different approximations to the rank one updates shown in these slides. For example, hierarchical decompositions.

These changes require altering how the T matrices are constructed, and will probably not have the same Galerkin condition as before.

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 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.
- 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

Schwarz as a fixed point iteration

Schwarz methods can be represented as fixed point iterations:

$$u^{n+1} = g\left(u^{n-1}\right),$$

where $g(u)$ represents one iteration of a Schwarz method that takes boundary data from the input u.

The fixed point of g is the solution on the first subdomain. This is also the root of the function $f(u) = g(u) - u$. We apply Newton's method:

$$u^* = u^{n-1} - J(u^{n-1})^{-1} f(u^{n-1}),$$

where J is the Jacobian of the function f.

Schwarz-Preconditioned Newton methods (Cai & Keyes, 2001)

This is equivalent to preconditioning Newton's method with a Schwarz method.

Let

$$f(u) = \begin{bmatrix} f_1(u_1, u_2) \\ f_2(u_1, u_2) \end{bmatrix}.$$

Then Newton's method tells us to solve the nonlinear problem

$$\begin{bmatrix} J_{11}^{(n+1)} & J_{12}^{(n+1)} \\ J_{21}^{(n+1)} & J_{22}^{(n+1)} \end{bmatrix} \begin{bmatrix} u_1^{n+1} - u_1^n \\ u_2^{n+1} - u_2^n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix},$$

where $J_{ij}^{(n+1)}$ is the derivative of f_i with respect to u_j evaluated at u_1^{n+1} and u_2^{n+1}.

Schwarz-Preconditioned Newton methods (Cai & Keyes, 2001)

Preconditioning this with a parallel Schwarz method tells us to solve

$$\begin{bmatrix} J_{11}^{(n+1)} & \\ & J_{22}^{(n+1)} \end{bmatrix} \begin{bmatrix} u_1^{n+1} - u_1^n \\ u_2^{n+1} - u_2^n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} - \begin{bmatrix} & J_{12}^{(n)} \\ J_{21}^{(n)} & \end{bmatrix} \begin{bmatrix} u_1^n - u_1^{n-1} \\ u_2^n - u_2^{n-1} \end{bmatrix},$$

and completing the iteration by solving

$$\begin{bmatrix} J_{11}^{(n+1)} & \\ & J_{22}^{(n+1)} \end{bmatrix}^{-1} \begin{bmatrix} J_{11}^{(n+1)} & J_{12}^{(n+1)} \\ J_{21}^{(n+1)} & J_{22}^{(n+1)} \end{bmatrix} \begin{bmatrix} u_1^* - u_1^n \\ u_2^* - u_2^n \end{bmatrix} = \begin{bmatrix} f_1(u_1^{n+1}, u_2^{n+1}) \\ f_2(u_1^{n+1}, u_2^{n+1}) \end{bmatrix}.$$

Alternate minimization in phase-field fracture models

Brittle fracture

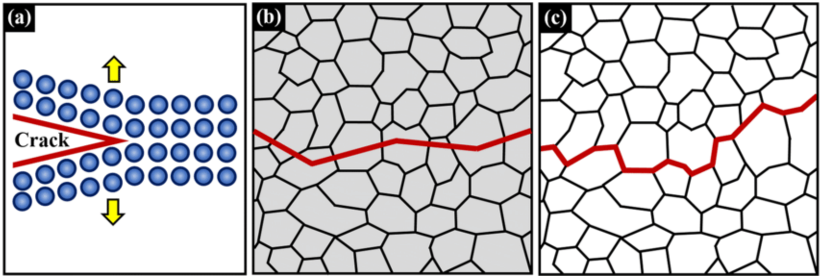


Figure 2: Crack propagation in brittle material, image taken from ResearchGate, uploaded by Dana Ashkenazi

Phase-field fracture model

$$E_\ell(u,\alpha) = \int \frac{1}{2}(1-\alpha)^2 Ae(u)\cdot e(u)dx + \frac{G_c}{4c_w}\int \frac{w(\alpha)}{\ell} + \ell\,|\nabla\alpha|^2\,dx$$

- u is a vector field which defines the displacement
- α is a scalar field that is 0 away from the crack and 1 on the crack
- $e(u)$ is the strain tensor, A the stiffness tensor
- ℓ determines the accuracy of the approximation of the Hausdorff measure of the crack
- $w(0)=0$, $w(1)=1$, $w'(x)\geq 0$, $w \in C^1(0,1)$, $c_w = \int_0^1 \sqrt{w(s)}ds$
- G_c is the critical energy release rate

Alternate minimization

Cracks propagate to minimize $E_\ell(u,\alpha)$. To model this, we then need to minimize over both u and α.

The energy E_ℓ is not convex in both variables, but it is convex if either u or α is kept constant. This leads to the idea of alternate minimization.

Algorithm 3 Alternate Minimization (AltMin)

```
1: Make initial guess $\alpha_0$ and set $n = 0$
2: while $\|\alpha_{n+1} - \alpha_n\|$ is greater than tolerance do
3:     Find $u_{n+1} = \operatorname{argmin}_u E_\ell(u, \alpha_n)$
4:     Find $\alpha_{n+1} = \operatorname{argmin}_\alpha E_\ell(u_{n+1}, \alpha)$
5: end while
```

Applying SPN (Kopanicakova, Kothari & Krause, 2023)

AltMin is in fact an alternating Schwarz method. It is then a natural candidate for SPN methods, which can be achieved by adding a Newton step after line 4:

$$\begin{bmatrix} J_{uu} & \\ J_{\alpha u} & J_{\alpha\alpha} \end{bmatrix}^{-1} \begin{bmatrix} J_{uu} & J_{u\alpha} \\ J_{\alpha u} & J_{\alpha\alpha} \end{bmatrix} \begin{bmatrix} u_* - u_n \\ \alpha_* - \alpha_n \end{bmatrix} = \begin{bmatrix} u_{n+1} - u_n \\ \alpha_{n+1} - \alpha_n \end{bmatrix},$$

where J_{ij} is the second derivative of $E_\ell(u, \alpha)$ with respect to i then j.

Issues with MSPIN on phase-field models

AltMin slows down significantly when it approaches local minima and saddle points. Away from these, both AltMin and MSPIN work great, so we need MSPIN to pick up the slack in these problem areas.

However, the Newton step becomes unstable in these areas. This can slow it down or even cause it to diverge.

We need some way to combine the AltMin step and Newton step to speed up AltMin in these problem areas.

Candidates:

- Anderson mixing using exact Jacobian
- Some modified Powell's dog-leg method