Adaptive optimized Schwarz methods

Conor McCoid

January 24th, 2025

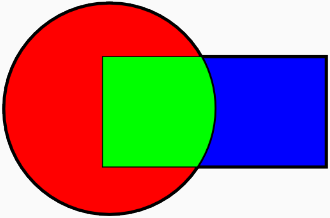
McMaster University

Brief history of 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:

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

P.L. Lions, 1989

Parallel Schwarz method:

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

Multiple subdomains:

$$\begin{cases} \Delta u_i^{n+1} = 0 & \text{in } \Omega_i, \\ u_i^{n+1} = u_j^n & \text{on } \Gamma_{ij} \end{cases}$$

where Γ_{ij} is the boundary of Ω_i that lies in Ω_j.

Algebraic domain decomposition

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

The domain Ω is split into Ω_1, which contains the variables in $\boldsymbol{u}_1$ and $\boldsymbol{u}_\Gamma$, and Ω_2, which contains the variables in $\boldsymbol{u}_2$ and $\boldsymbol{u}_\Gamma$.

The variables $\boldsymbol{u}_\Gamma$ are shared between the two subdomains and forms an **interface** between the two.

Algebraic Schwarz methods

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.

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.

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

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

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

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

$V_n W_n^\top \approx E$

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

Choice of adaptive transmission conditions

For our adaptive transmission conditions, we want

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

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

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

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 $i = 1:2$ **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 $\boldsymbol{d}^{n+1}_j$

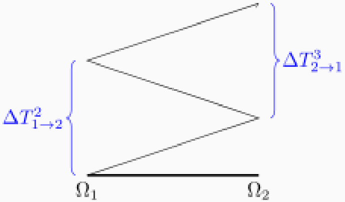
        $\boldsymbol{u}^{n+1}_j := \boldsymbol{u}^{n-1}_j + \boldsymbol{d}^{n+1}_j$, $\boldsymbol{u}^{n+1}_{j\Gamma} := \boldsymbol{u}^{n-1}_{j\Gamma} + \boldsymbol{d}^{n+1}_{j\Gamma}$

        $n \leftarrow n + 1$

    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

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

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

Numerical results

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

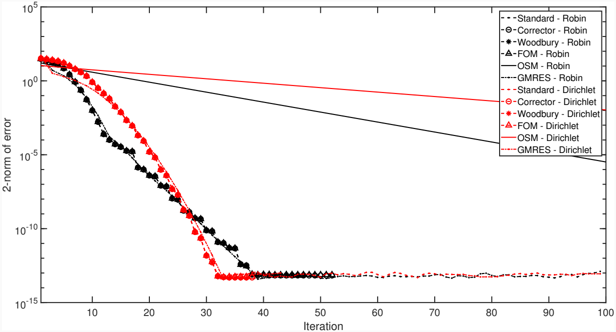


Figure 1: Comparison of altAOSM versions with $N = 10,000$, $M = 100$, using both Robin and Dirichlet boundary conditions.

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.
- Nested decompositions will allow more subdomains, but crosspoints need to be resolved for more general decompositions.