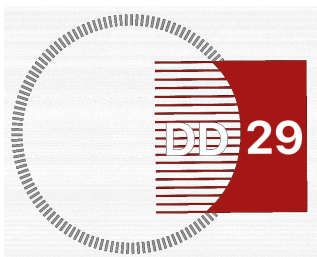
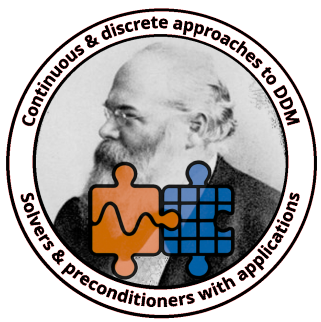
Symmetrized cells in adaptive optimized Schwarz

Conor McCoid

June 24th, 2025

McMaster University





Domain decomposition

What is domain decomposition?

Domain decomposition splits large complicated domains into smaller, easier subdomains. The problem is solved on each subdomain, information is passed to other subdomains, then solved on subdomains again, etc., creating an iterative method.

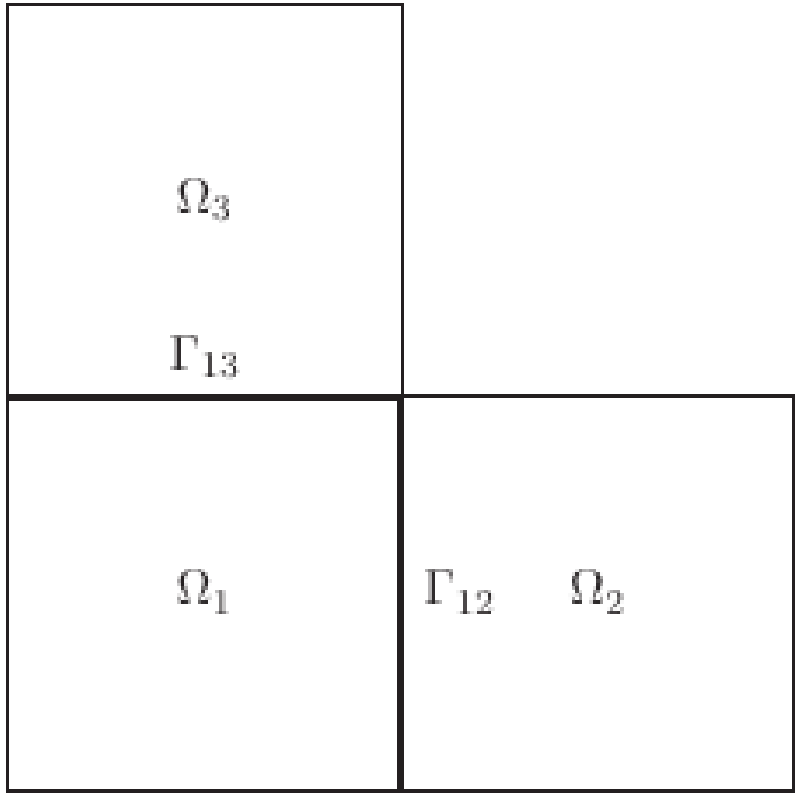


Figure 1: Example taken from "Iterative Methods for Sparse Linear Systems", by Yousef Saad

Discretization of the example

Now that we have our continuous domain decomposition, we need to discretize the problem and split up the unknowns.

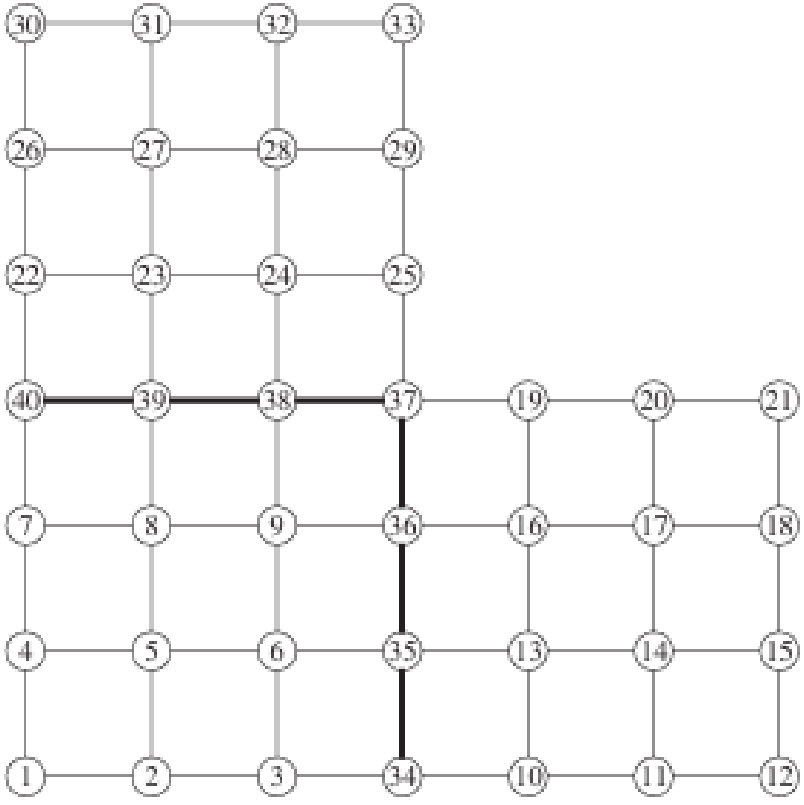


Figure 2: Nodes for example

Matrix of the example

This discretization, with this ordering for the nodes, comes with a matrix, with one row and one column per node. The dashed lines show how the domain decomposition splits up the unknowns.

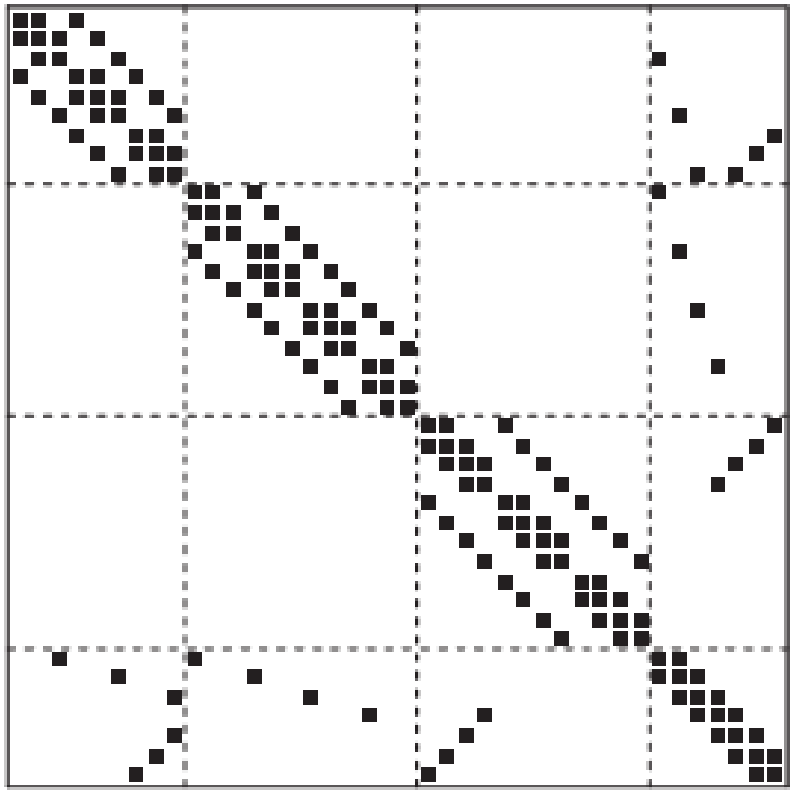


Figure 3: Matrix for example based on nodes

More complicated example

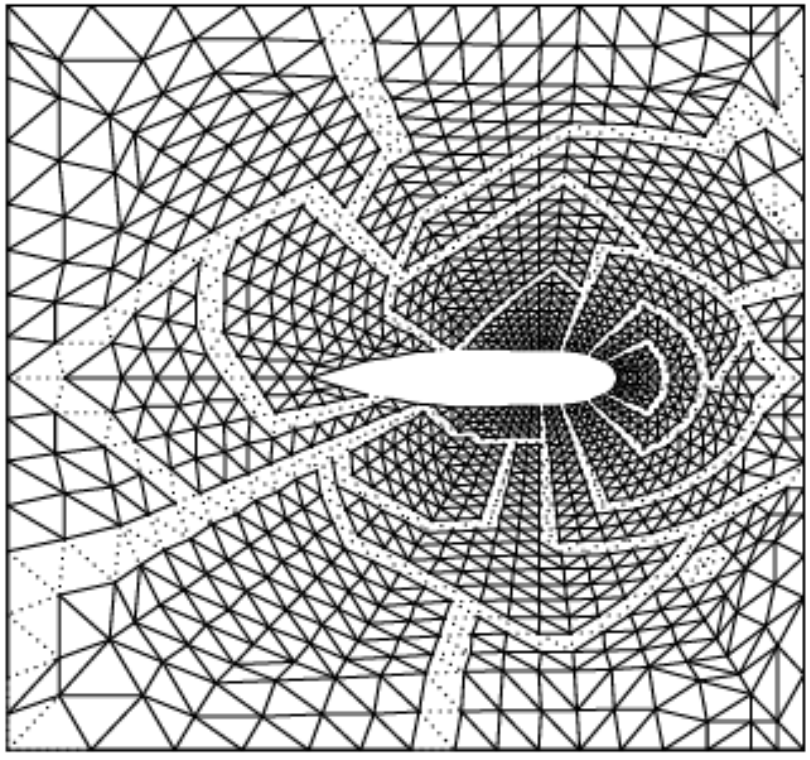


Figure 4: Also from Saad's "Iterative Methods"

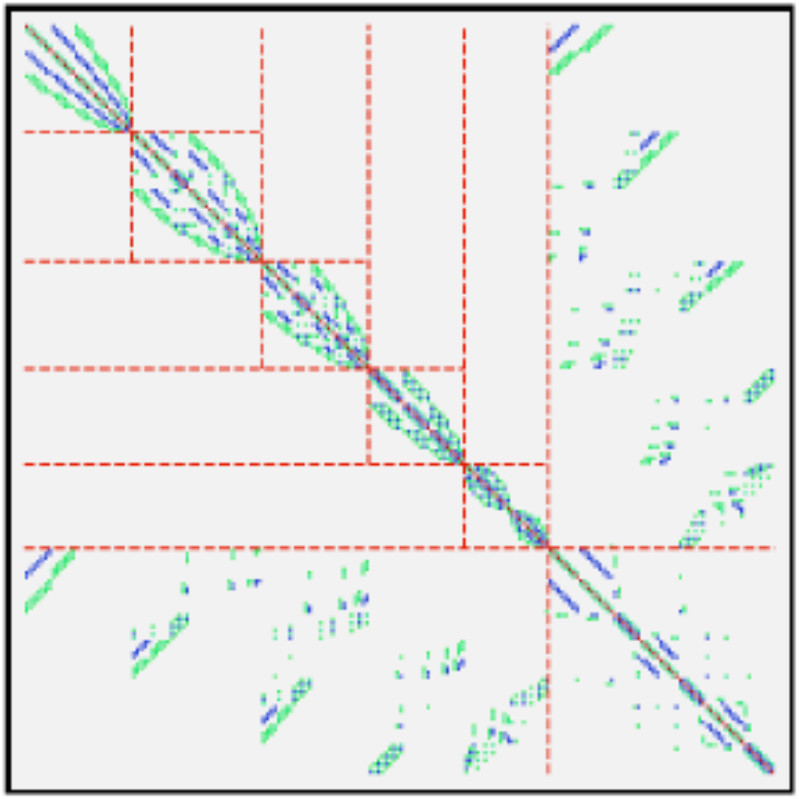


Figure 5: Possible matrix for example

A general form of the matrix

In this talk, let us consider matrices that can take the form

$$\begin{bmatrix} A_{11} & & & & A_{1\Gamma} \\ & A_{22} & & & A_{2\Gamma} \\ & & \ddots & & \vdots \\ & & & A_{nn} & A_{n\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & \dots & A_{\Gamma n} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_n \\ \boldsymbol{u}_\Gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_n \\ \boldsymbol{f}_\Gamma \end{bmatrix}, \tag{1}$$

where A_{ii} are square. This system represents n subdomains connected through a global interface represented by Γ.

The subproblems

Each subdomain now has its own subproblem:

$$\begin{bmatrix} A_{ii} & A_{i\Gamma} \\ A_{\Gamma i} & A_{\Gamma\Gamma} + S_i \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_\Gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \tilde{\boldsymbol{f}}_i \end{bmatrix}, \tag{2}$$

where $\tilde{\boldsymbol{f}}_i$ is some modification of $\boldsymbol{f}_\Gamma$, and S_i is some global transmission matrix.

We see that the variables associated with the global interface Γ appear in all subproblems. The global transmission matrix S_i dictates how these many copies communicate with one another.

How to choose S_i and $\tilde{f}_i$

There are perfect choices of S_i and $\tilde{\boldsymbol{f}}_i$ such that each subproblem gives the exact solution to the global problem on its respective subdomain. However, these perfect choices are expensive to compute.

Instead, the standard procedure is to make *a priori* choices that give convergent iterative methods. These appear as:

$$\begin{bmatrix} A_{ii} & A_{i\Gamma} \\ A_{\Gamma i} & A_{\Gamma\Gamma} + S_i \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(k+1)} \\ \boldsymbol{u}_{\Gamma i}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_\Gamma \end{bmatrix} + \sum_{j \neq i} \begin{bmatrix} & \\ -A_{\Gamma j} & T_j \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_j^{(k)} \\ \boldsymbol{u}_{\Gamma j}^{(k)} \end{bmatrix}, \tag{3}$$

where

$$S_i = \sum_{j \neq i} T_j. \tag{4}$$

Choices for T_j

The local transmission matrices T_j can represent boundary conditions between the subdomains. Some common options:

- Dirichlet, setting the interface variables on subdomain 2 equal to those found on subdomain 1
- Neumann, setting the derivatives to be the same
- Optimized, setting Robin boundary conditions to be the same, using a Robin parameter that optimizes convergence rates

In this talk, there won't be an *a priori* choice. The transmission matrices will adapt to the iterative method.

Symmetrized cells

What is a symmetrized cell?

For each subdomain, take a copy of it and stitch it together along their shared interface. This pair is now perfectly symmetric, and one subproblem describes both copies.

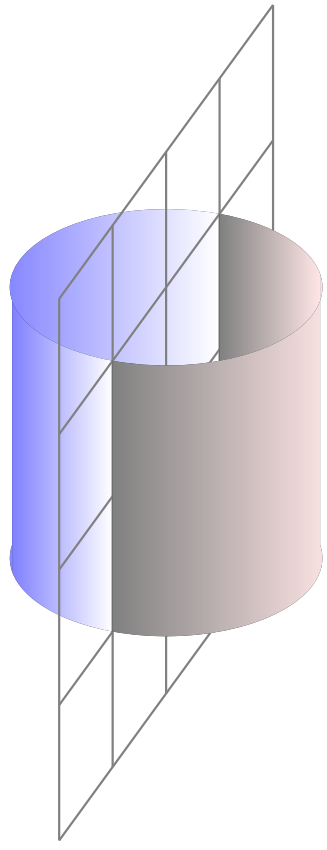


Figure 6: A symmetrized square domain with interfaces on two opposing edges

Algebra of a symmetrized cell

The equation for the unknowns on a symmetrized cell is:

$$\begin{bmatrix} A_{ii} & A_{i\Gamma} & \\ A_{\Gamma i} & A_{\Gamma\Gamma} & A_{\Gamma i} \\ & A_{i\Gamma} & A_{ii} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}}_i \\ \hat{\boldsymbol{u}}_\Gamma \\ \hat{\boldsymbol{u}}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \hat{\boldsymbol{f}}_i \\ \boldsymbol{f}_i \end{bmatrix}. \tag{5}$$

The solution on this cell does not correspond to the solution of the global problem, but it may serve as a good initial guess.

Subproblems of the cell

The iterative process of solving the equation on the cell is then solving repeatedly the equation:

$$\begin{bmatrix} A_{ii} & A_{i\Gamma} \\ A_{\Gamma i} & A_{\Gamma\Gamma} + T_i^{(k+1)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(k+1)} \\ \boldsymbol{u}_{\Gamma i}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \hat{\boldsymbol{f}}_i \end{bmatrix} + \begin{bmatrix} & \\ -A_{\Gamma i} & T_i^{(k)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(k)} \\ \boldsymbol{u}_{\Gamma i}^{(k)} \end{bmatrix}. \tag{6}$$

It is often better to present this in a corrector version that acts only on the differences between successive iterates, $\boldsymbol{d}_i^{(k+1)} = \boldsymbol{u}_i^{(k+1)} - \boldsymbol{u}_i^{(k)}$:

$$\begin{bmatrix} A_{ii} & A_{i\Gamma} \\ A_{\Gamma i} & A_{\Gamma\Gamma} + T_i^{(k+1)} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_i^{(k+1)} \\ \boldsymbol{d}_{\Gamma i}^{(k+1)} \end{bmatrix} = \begin{bmatrix} & \\ -A_{\Gamma i} & T_i^{(k)} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_i^{(k)} \\ \boldsymbol{d}_{\Gamma i}^{(k)} \end{bmatrix}. \tag{7}$$

Krylov subspace in the cell

We can use techniques from static condensation to reduce the form of this system to acting only on the global interface:

$$\boldsymbol{d}_{\Gamma i}^{(k+1)} = \left(\hat{A}_i + E_i^{(k+1)}\right)^{-1} E_i^{(k)} \boldsymbol{d}_{\Gamma i}^{(k)}, \tag{8}$$

where

$$\hat{A}_i = A_{\Gamma\Gamma} - 2A_{\Gamma i}A_{ii}^{-1}A_{i\Gamma}, \quad E_i^{(k)} = T_i^{(k)} + A_{\Gamma i}A_{ii}^{-1}A_{i\Gamma}.$$

This means the vectors $\boldsymbol{d}_{\Gamma i}^{(k+1)}$ form a Krylov subspace:

$$\boldsymbol{d}_{\Gamma i}^{(k+1)} \in \mathcal{K}_{k+1}\left(\left(\hat{A}_i + E_i^{(k+1)}\right)^{-1} E_i^{(k)}, \boldsymbol{d}_{\Gamma i}^{(0)}\right). \tag{9}$$

Updates to $T_i^{(k)}$

The most important part of the symmetrized cells is the update to the transmission matrix T_i. We want an update that uses information obtained in solving equations on the cell and that reduces $E_i^{(k)}$ to zero, as this means the iterative process becomes direct.

As a first approximation, use rank one updates that give the action of $E_i^{(k)}$ in the direction $\boldsymbol{d}_{\Gamma i}^{(k)}$:

$$T_i^{(k+1)} - T_i^{(k)} := -E_i^{(k)} \frac{\boldsymbol{d}_{\Gamma i}^{(k)} \left(\boldsymbol{d}_{\Gamma i}^{(k)}\right)^\top}{\left\| \boldsymbol{d}_{\Gamma i}^{(k)} \right\|^2}.$$

Iterative action approximation

If the vectors $\boldsymbol{d}_{\Gamma i}^{(k)}$ aren't orthogonal, then these updates will interfere with each other. Run these vectors through modified Gram-Schmidt to fix this issue.

1: Inputs: $\boldsymbol{d}_{\Gamma i}^{(k)}$, $E_i^{(k)}\boldsymbol{d}_{\Gamma i}^{(k)}$, all previous $\boldsymbol{d}_{\Gamma i}^{(j)}$ and $E_i^{(j)}\boldsymbol{d}_{\Gamma i}^{(j)}$
2: Set $\boldsymbol{w}_k := \boldsymbol{d}_{\Gamma i}^{(k)}$ and $\boldsymbol{v}_k := E_i^{(k)}\boldsymbol{d}_{\Gamma i}^{(k)}$
3: **for** $j = 1 : k-1$ **do**
4: $\quad h \leftarrow \langle \boldsymbol{d}_{\Gamma i}^{(j)}, \boldsymbol{w}_k \rangle$, $\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k - h\boldsymbol{d}_{\Gamma i}^{(j)}$
5: $\quad \boldsymbol{v}_k \leftarrow \boldsymbol{v}_k - hE_i^{(j)}\boldsymbol{d}_{\Gamma i}^{(j)}$
6: **end for**
7: Output: $\Delta T_i^{(k+1)} := -\boldsymbol{v}_k \boldsymbol{w}_k^\top$

Krylov subspace method

Incidentally, this will give an orthonormal basis for the Krylov subspace. There is also an indirect Galerkin condition at work, making this process essentially equivalent to the full orthogonalization method (FOM), precursor to GMRES.

To show that, use the Woodbury matrix identity to write $\boldsymbol{d}_{\Gamma i}^{(k+1)}$ as

$$\boldsymbol{d}_{\Gamma i}^{(k+1)} = \frac{\left(\hat{A}_i + E_i^{(k)}\right)^{-1} E_i^{(k)} \boldsymbol{d}_{\Gamma i}^{(k)}}{\boldsymbol{w}_k^\top \left(I - \left(\hat{A}_i + E_i^{(k)}\right)^{-1} E_i^{(k)}\right) \boldsymbol{w}_k}$$

Indirect Galerkin condition

Next, consider the following Galerkin condition:

$$\left(I - \left(\hat{A}_i + E_i^{(k)}\right)^{-1} E_i^{(k)}\right) \boldsymbol{x} - \boldsymbol{d}_{\Gamma i}^{(k)} \perp \mathcal{K}_k, \quad \boldsymbol{x} \in \mathcal{K}_k.$$

The solution satisfies

$$\boldsymbol{w}_k^\top \boldsymbol{x} = \frac{\boldsymbol{w}_k^\top \boldsymbol{d}_{\Gamma i}^{(k)}}{\boldsymbol{w}_k^\top \left(I - \left(\hat{A}_i + E_i^{(k)}\right)^{-1} E_i^{(k)}\right) \boldsymbol{w}_k}.$$

Since $E_i^{(k)}$ has $\mathcal{K}_{k-1}$ in its null space, we can write $\boldsymbol{d}_{\Gamma i}^{(k+1)}$ as

$$\boldsymbol{d}_{\Gamma i}^{(k+1)} = \left(\hat{A}_i + E_i^{(k)}\right)^{-1} E_i^{(k)} \boldsymbol{x}.$$

Results of symmetrized cells

Once the iterative process on the cell is complete, we get the following outputs:

excellent local transmission matrix: the matrix T_i (last to be computed) is now a very good approximation of the best possible local transmission matrix;

good initial guess of the solution on the cell: the solution $\hat{u}_i$ is not the solution for the global problem, but it satisfies the problem locally, making it a good initial guess for the next stage.

Continuous analogs

The matrix T_i approximates absorbing boundary conditions. These conditions allow waves to pass through the boundary with no reflection.

Several methods in the continuous space have been developed to approximate ABCs:

- perfectly matched layers
- one-way equations
- change of variables
- probing

Global reconstruction

Putting the pieces together again

Now that we have good transmission matrices and good initial guesses, we can solve the global problem very quickly. Set

$$S_i = \sum_{j \neq i} T_j, \quad \boldsymbol{u}_i^{(0)} = \hat{\boldsymbol{u}}_i,$$

then solve

$$\begin{bmatrix} A_{ii} & A_{i\Gamma} \\ A_{\Gamma i} & A_{\Gamma\Gamma} + S_i \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(k+1)} \\ \boldsymbol{u}_{\Gamma i}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_\Gamma \end{bmatrix} + \sum_{j \neq i} \begin{bmatrix} & \\ -A_{\Gamma j} & T_j \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_j^{(k)} \\ \boldsymbol{u}_{\Gamma j}^{(k)} \end{bmatrix},$$

or a corrector version if preferred:

$$\begin{bmatrix} A_{ii} & A_{i\Gamma} \\ A_{\Gamma i} & A_{\Gamma\Gamma} + S_i \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_i^{(k+1)} \\ \boldsymbol{d}_{\Gamma i}^{(k+1)} \end{bmatrix} = \sum_{j \neq i} \begin{bmatrix} & \\ -A_{\Gamma j} & T_j \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_j^{(k)} \\ \boldsymbol{d}_{\Gamma j}^{(k)} \end{bmatrix}.$$

Condensed equation

Using the corrector version and the same techniques from static condensation, the iteration can be represented acting solely on the global interface:

$$\left(\hat{A} + \sum_{j \neq i} E_j\right) \boldsymbol{d}_{\Gamma i}^{(k+1)} = \sum_{j \neq i} E_j \boldsymbol{d}_{\Gamma j}^{(k)},$$

where

$$\hat{A} = A_{\Gamma\Gamma} - \sum_{i=1}^{n} A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma}, \quad E_j = T_j + A_{\Gamma j} A_{jj}^{-1} A_{j\Gamma}.$$

Convergence rate

This means the convergence of the global reconstruction is defined by

$$\begin{bmatrix} \vdots \\ \boldsymbol{d}_{\Gamma i}^{(k+1)} \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & & \\ & \hat{A} + \sum_{j \neq i} E_j & \\ & & \ddots \end{bmatrix}^{-1} \begin{bmatrix} & E_2 & E_3 & \cdots \\ E_1 & & E_3 & \cdots \\ E_1 & E_2 & & \\ \vdots & \vdots & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \boldsymbol{d}_{\Gamma i}^{(k)} \\ \vdots \end{bmatrix}.$$

The spectral radius of the matrix in this equation then dictates the convergence rate. After the symmetrized cells iterations, this spectral radius should be small, but more analysis is needed to confirm this in general.

Rectangular domain split into strips

The test problem used is the Poisson equation on a rectangular domain. The subdomains are squares. In weak scaling, we increase both the size of the domain and the number of strips. In strong scaling, we keep the domain fixed but increase the number of strips.

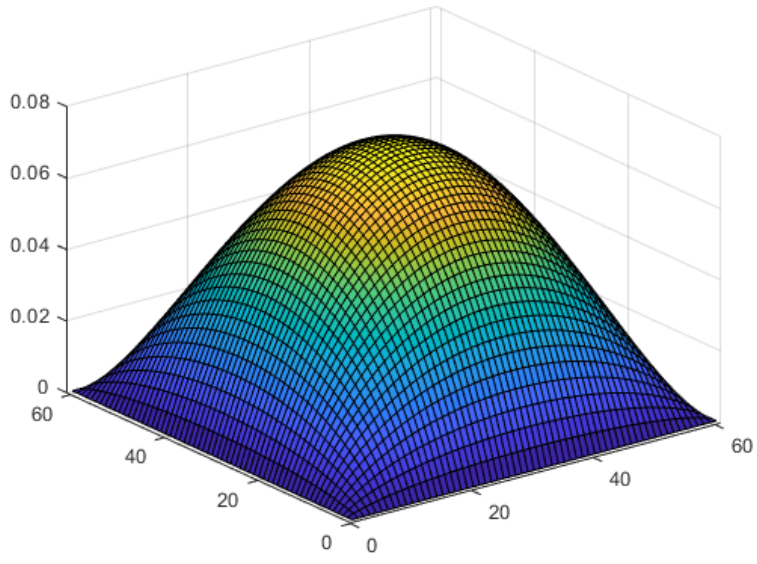


Figure 7: Solution on square domain

Weak scaling

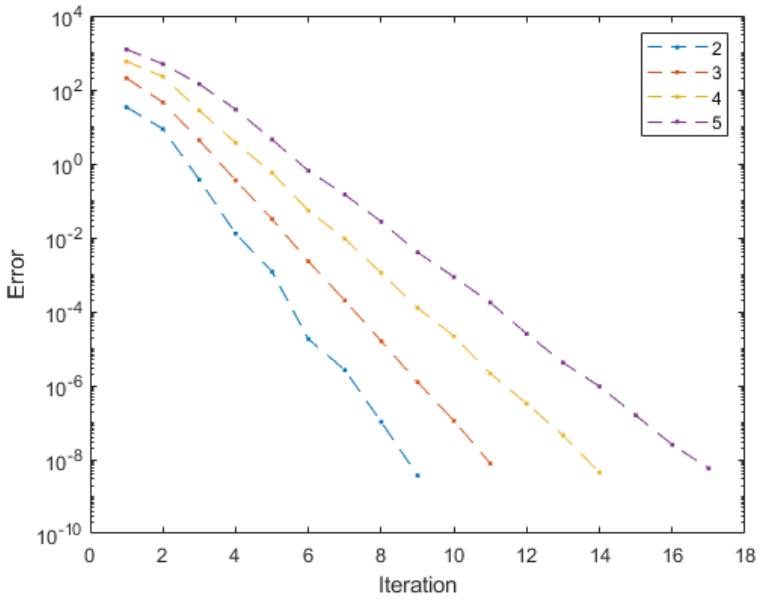


Figure 8: Error in global reconstruction

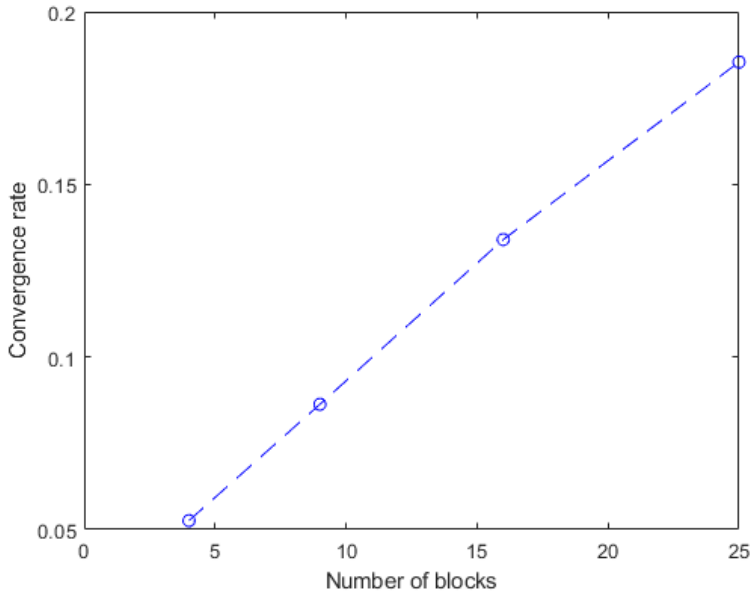


Figure 9: Convergence rate by number of blocks

Strong scaling

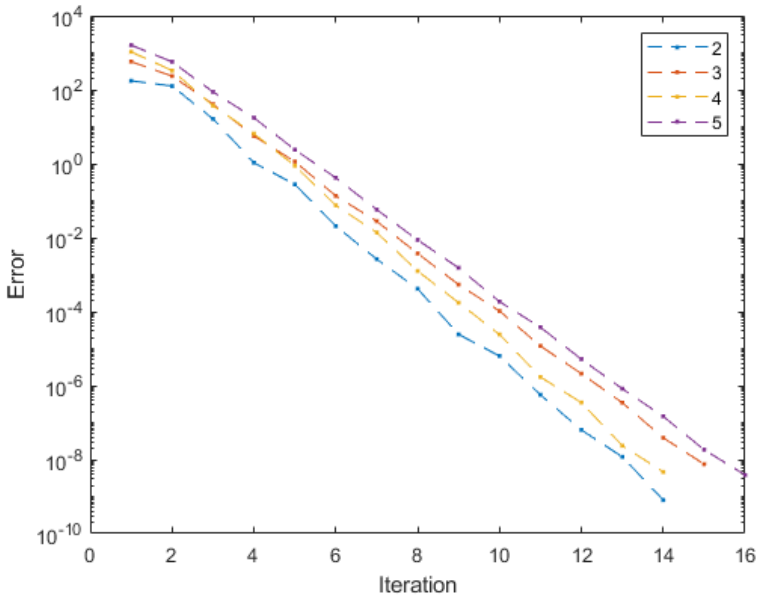


Figure 10: Error in global reconstruction

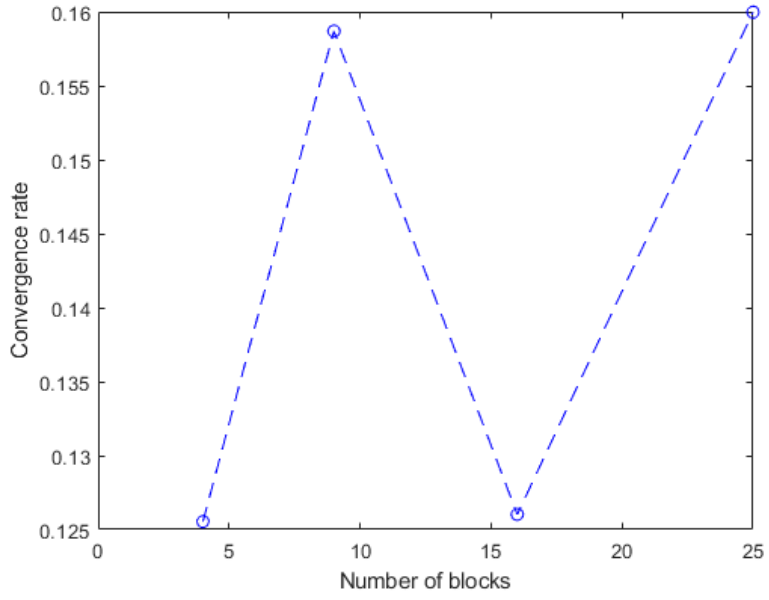


Figure 11: Convergence rate by number of blocks

Conclusions and future work

- Symmetrized cells is a highly parallelizable method for constructing highly convergent transmission matrices
- Global reconstruction is scalable for some simple, small examples
- Scalability needs to be tested on HPC examples with a parallelized code base

FLOP count

k maximum number of symmetrized cell iterations

ℓ number of global reconstruction iterations

m number of cells

N size of each cell

M size of the global interface

	Symmetrized cells	Global reconstruction
mult.	$mkM(N+M) + m\frac{3k^2}{2}M$	$m\ell M(N+M)$
add.	$mkM(N+M) + m\frac{3k^2}{2}M$	$m\ell(N+M) + mM$
sys. solve	mk	$m\ell$