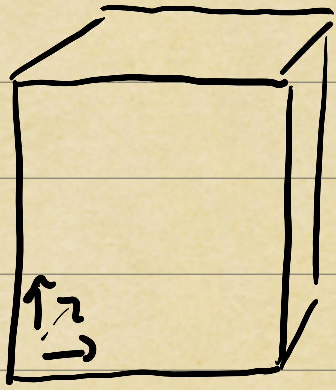
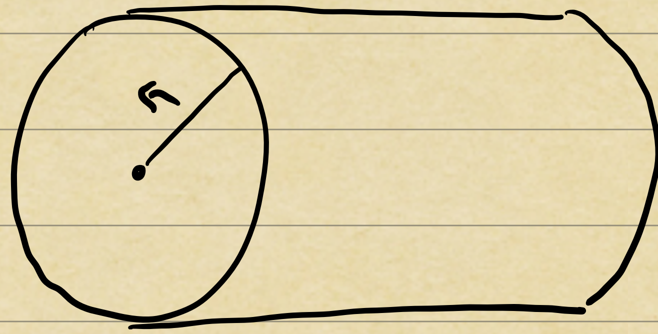
Domain Decomposition

On separable domains, we can often build fast 2 or 3D solvers from 1D building blocks.

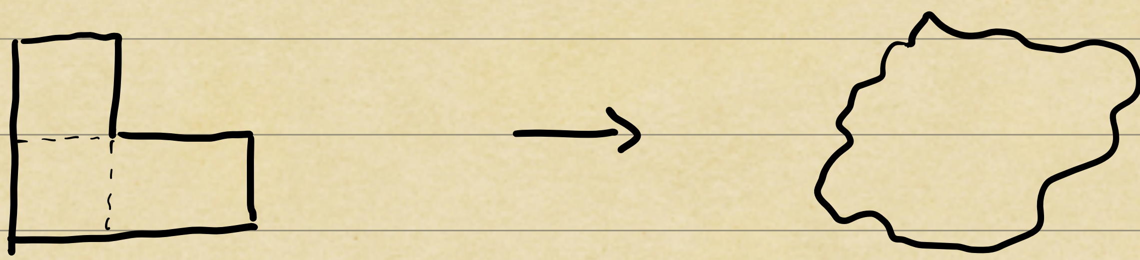


$[0\ 1] \times [0\ 1] \times [0\ 1]$



$[0,1] \times [0\ 1] \times [0\ 1)$ (periodic)

What about more complicated geometries?



"simple" but nonseparable

"complex" and nonseparable

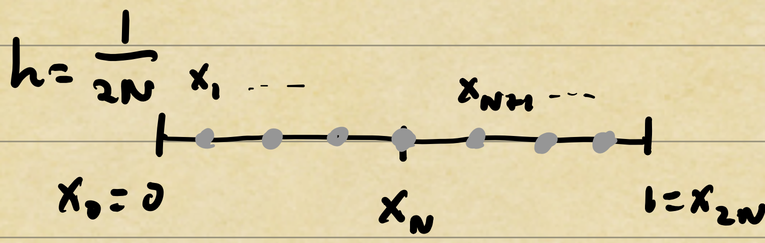
Domain Decomposition splits the domain into a collection of simple geometric domains and carefully "glues" together solutions on each piece.

Schur Complement/Direct Substructuring Method

To illustrate the basic ideas, start in 1D.

$-u_{xx} = f$ on $[0,1]$

with $u(0) = u(1) = 0$



$\Rightarrow$ Solve on two smaller "patches" $[0, \frac{1}{2}]$ and $[\frac{1}{2}, 1]$, and then "glue" solutions together.

$\Rightarrow$ Need to find $u_N \approx u(x_N)$ to solve the two smaller Poisson problems.

Write down 2^{nd} order central FD discretization:

$$
\left[\begin{array}{cccc|cccc|c}
2 & -1 & & & & & & & 0 \\
-1 & 2 & \ddots & & & & & & \vdots \\
 & \ddots & \ddots & -1 & & & & & \vdots \\
 & & -1 & 2 & & & & & -1 \\
\hline
 & & & & 2 & -1 & & & -1 \\
 & & & & -1 & 2 & \ddots & & 0 \\
 & & & & & \ddots & \ddots & -1 & \vdots \\
 & & & & & & -1 & 2 & 0 \\
\hline
0 & \cdots & 0 & -1 & -1 & 0 & \cdots & 0 & 2
\end{array}\right]
\left[\begin{array}{c}
u_1 \\ \vdots \\ \vdots \\ u_{N-1} \\ \hline u_{N+1} \\ \vdots \\ \vdots \\ u_{2N-1} \\ \hline u_N
\end{array}\right]
=
\left[\begin{array}{c}
f_1 \\ \vdots \\ \vdots \\ f_{N-1} \\ \hline f_{N+1} \\ \vdots \\ \vdots \\ f_{2N-1} \\ \hline f_N
\end{array}\right]
$$

re-order to isolate DOF on "glue" (x_N)

The re-ordered linear system has the form:

$$\begin{array}{cc} & \begin{array}{ccc}(N-1)\times(N-1) & \qquad\qquad & (N-1)\times 1\end{array} \\ \begin{array}{c} \\ \\ \\ 1\times(N-1)\end{array} & \left[\begin{array}{c|c|c} A_{11} & & A_{1\Gamma} \\ \hline & A_{22} & A_{2\Gamma} \\ \hline A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma}\end{array}\right]\end{array} \left[\begin{array}{c}\underline{u}_1\\ \hline \underline{u}_2 \\ \hline u_\Gamma\end{array}\right] = \left[\begin{array}{c}\underline{f}_1\\ \hline \underline{f}_2\\ \hline f_\Gamma\end{array}\right]$$

The two $(N-1)\times(N-1)$ diagonal blocks correspond to Poisson problems on the left & right patches, while the last row & col correspond to interactions between the patches and the glue.

Can we solve for u_N w/out factoring the entire matrix (solving for $u_1, \ldots, u_{2N}$)?

$\Rightarrow$ Block elimination/Schur Complement

<u>Step 1:</u>

$$\begin{bmatrix} I & & A_{11}^{-1}A_{1\Gamma} \\ & I & A_{22}^{-1}A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma}\end{bmatrix}\begin{bmatrix}\underline{u}_1\\ \underline{u}_2\\ u_\Gamma\end{bmatrix} = \begin{bmatrix}A_{11}^{-1}\underline{f}_1\\ A_{22}^{-1}\underline{f}_2\\ f_\Gamma\end{bmatrix}$$

Step 2:

$$\begin{bmatrix} A_{\Gamma 1} & & A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma} \\ & A_{\Gamma 2} & A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \underline{u}_1 \\ \underline{u}_2 \\ u_\Gamma \end{bmatrix} = \begin{bmatrix} A_{\Gamma 1}A_{11}^{-1}\underline{f}_1 \\ A_{\Gamma 2}A_{22}^{-1}\underline{f}_2 \\ f_\Gamma \end{bmatrix}$$

Step 3: Subtract rows 1 & 2 from row 3 to eliminate all but last entry of last row.

$$\Rightarrow \left(A_{\Gamma\Gamma} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma}\right)u_\Gamma = f_\Gamma - A_{\Gamma 1}A_{11}^{-1}\underline{f}_1 - A_{\Gamma 2}A_{22}^{-1}\underline{f}_2$$

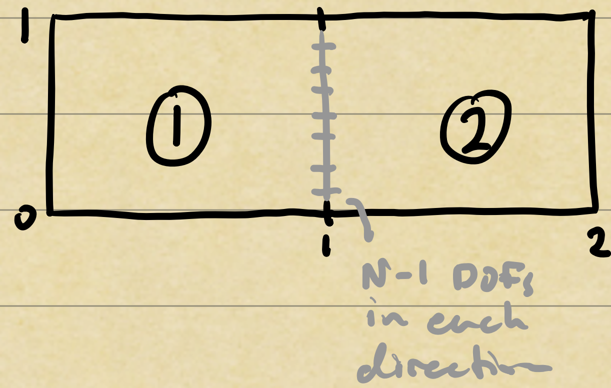
Now have a 1x1 eqn for u_Γ, after which we can solve the two separate patches separately.

$\Rightarrow$ Need 2 solves w/ A_{11} and 2 solves w/ A_{22} to compute left- and right-hand sides.

Once we have u_N, just solve 2 Poisson problems of half the size w/ Dirichlet data determined by $u_0 = 0$, $u_{2N} = 0$, and $u_N = 0$.

Complexity boils down to complexity of the Poisson solves on each patch.

How does the process look in 2D?



$$-u_{xx} - u_{yy} = f$$

$$u\big|_{\partial\Omega} = 0$$

System has same structure, but different dims

$$\begin{array}{c c} & \begin{array}{ccc} (N-1)^2 & (N-1)^2 & N-1 \end{array} \\ \begin{array}{c} (N-1)^2 \\ (N-1)^2 \\ N-1 \end{array} & \begin{bmatrix} A_{11} & & A_{1\Gamma} \\ & A_{22} & A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma} \end{bmatrix} \end{array} \begin{bmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \underline{u}_\Gamma \end{bmatrix} = \begin{bmatrix} \underline{f}_1 \\ \underline{f}_2 \\ \underline{f}_\Gamma \end{bmatrix}$$

where $\quad A_{11} = A_{22} = \frac{1}{h^2}(K \otimes I + I \otimes K)$

(2D Poisson w/ Dirichlet B.C.s)

<u>Schur Complement:</u>

$$\underbrace{\left(A_{\Gamma\Gamma} - A_{\Gamma 1} A_{11}^{-1} A_{1\Gamma} - A_{\Gamma 2} A_{22}^{-1} A_{2\Gamma}\right)}_{S} \underline{u}_\Gamma = \underline{f}_\Gamma - A_{\Gamma 1} A_{11}^{-1} \underline{f}_1 - A_{\Gamma 2} A_{22}^{-1} \underline{f}_2$$

S is now an $(N-1)\times(N-1)$ matrix w/ generically dense structure.

Forming S naively requires $(N-1)$ Poisson solves on each patch, and then a dense solve to compute u_Γ, the boundary data.

$$\Rightarrow \mathcal{O}(N^3 \log N)$$

However, one can apply S to a vector in $\mathcal{O}(N^2 \log N)$ time, so one might consider an iterative solver for u_Γ.