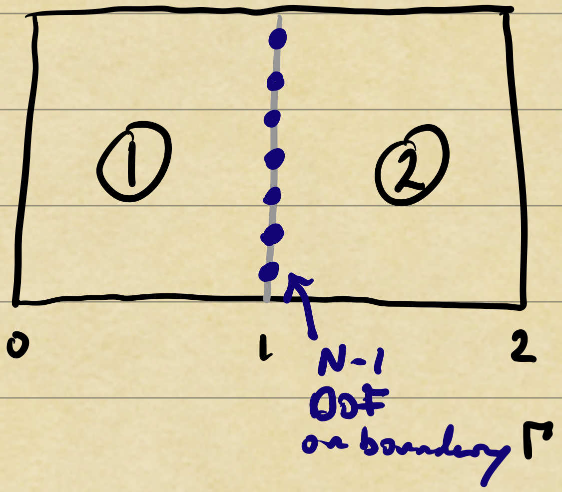
Domain Decomposition (pt 2)



$-\Delta u = f$ with $u|_{\partial\Omega} = 0$

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

Schur Complement: $\quad S\underline{u}_\Gamma = \tilde{\underline{f}}_\Gamma$

for N-1 boundary vals

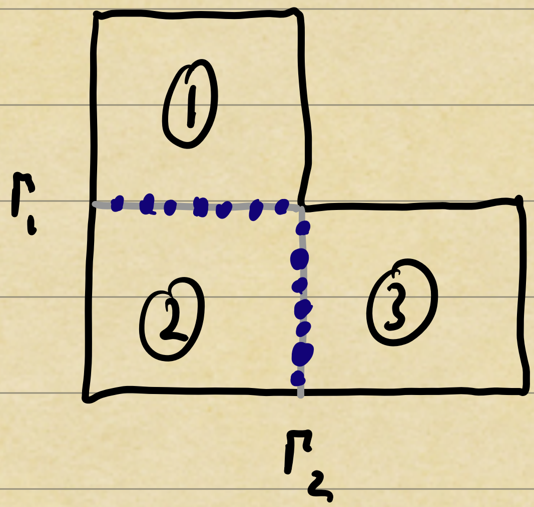
$$S\underline{u}_\Gamma = \big(A_{\Gamma\Gamma} - \underbrace{A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma}}_{2(N-1)\text{ fast solves}}\big)\underline{u}_\Gamma$$

$\Rightarrow$ Even with $\mathcal{O}(N^2\log_2 N)$ solves on (1) & (2), S takes $\mathcal{O}(N^3\log_2 N)$ flops to form and solve

$\Rightarrow$ However, $x \to Sx$ takes only $\mathcal{O}(N^2\log_2 N)$

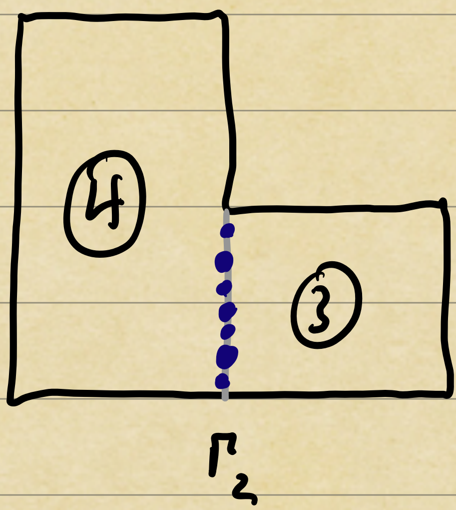
$\Rightarrow$ Iterative solve for u_Γ if S is well-conditioned or good preconditioner available.

Hierarchical Merging



$$\left[\begin{array}{ccc|c|c} A_{11} & & A_{1\Gamma_1} & & \\ & A_{22} & A_{2\Gamma_1} & & A_{2\Gamma_2} \\ A_{\Gamma_1 1} & A_{\Gamma_1 2} & A_{\Gamma_1\Gamma_1} & & \\ \hline & & & A_{33} & A_{3\Gamma_2} \\ \hline & A_{\Gamma_2 2} & & A_{\Gamma_2 3} & A_{\Gamma_2\Gamma_2} \end{array}\right]$$

$$\Downarrow$$

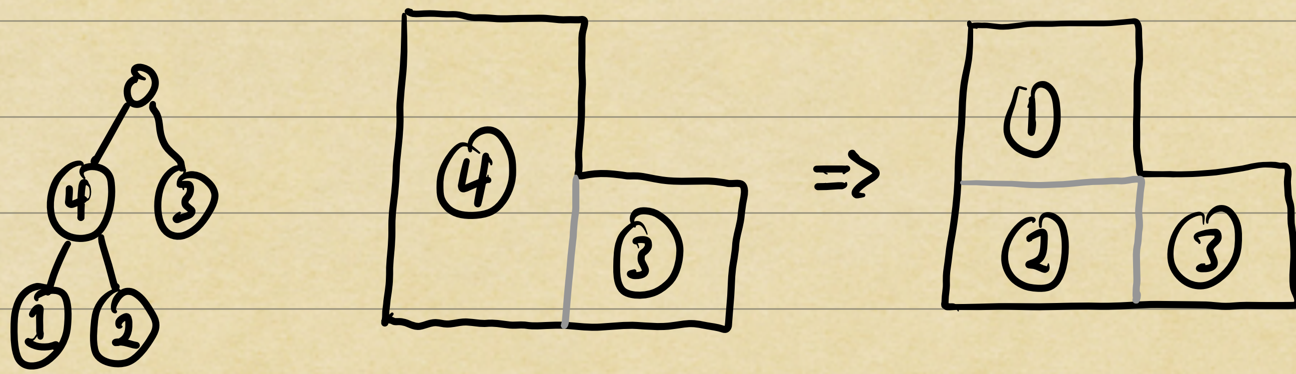


$$\left[\begin{array}{c|c|c} A_4 & & A_{2\Gamma_2} \\ \hline & A_{33} & A_{3\Gamma_2} \\ \hline A_{\Gamma_2 2} & A_{\Gamma_2 3} & A_{\Gamma_2\Gamma_2} \end{array}\right]$$

Same structure as original pairwise merge!

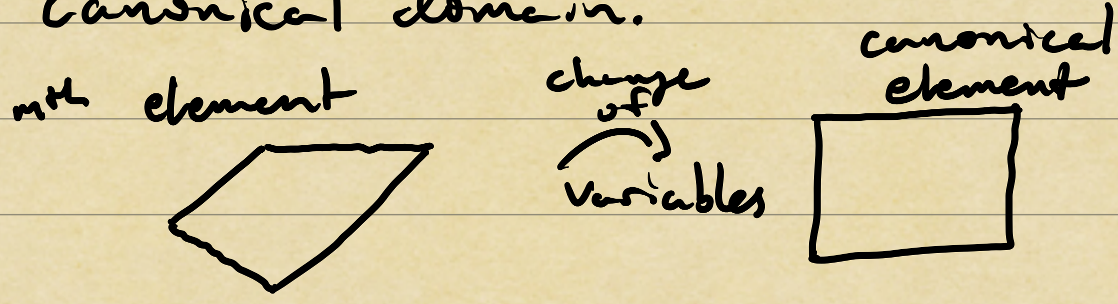
$\Rightarrow$ Schur Complement for unknowns on Γ_2 now uses fast solves on 3 and 4, where the fast solve on 4 is based on the fast pairwise merge that uses fast solvers on 1 and 2.

⇒ Recursive subdivision of domain and hierarchical pairwise merges exploit fast solvers on each patch for fast solver on the whole domain.



We will take a closer look at fast *direct* solvers based on hierarchical subdivision and merging later in the course.

⇒ Often take subdivision based on some looser collection of shapes (triangles, quadrilateral, etc.) that map back to canonical domain.



Then, PDE on each "element" goes through change-of-variables and the mod'd PDE can be solved in canonical domain.

Spectral Methods

So far we have been developing fast FD Poisson solvers based on FFTs, which essentially exploit the translation invariance of (const. coeffs) differential operators.

We've also discussed how to leverage fast solvers for simple 1D problems to tackle more complicated PDEs.

The next section of the course is about designing fast solvers that exploit smoothness in the PDE to achieve superior rates of convergence (for smooth problems) in the size of the discretization N.

FD methods use local fixed order (p) polynomial approximations to achieve

algebraic convergence: error $\sim N^{-p}$

e.g., when function has $p+1$ continuous derivatives.

Spectral methods use global polynomial (or more general function) approximations for

superalgebraic convergence: error $= \mathcal{O}(N^{-p})$ for any fixed order $p = 1, 2, 3, \ldots$

e.g. when function is infinitely differentiable.

$$\underset{\substack{\uparrow \\ \text{function} \\ \text{to approx.}}}{f(x)} \approx \sum_{k=0}^{N-1} \overset{\text{expansion coeffs}}{\tilde{f}_n} \underset{\substack{\text{approximation set,} \\ \text{e.g. polynomials}}}{\phi_n(x)}$$

$\Rightarrow$ Design fast solvers based on spectrally accurate function expansions.