

CME342/AA220/CS 238
Parallel Methods in Numerical Analysis

Domain Decomposition Methods II

Outline:

- Overlapping methods with many subdomains.
- Nonoverlapping methods.

Announcements

- HW#5 / Final Project due Wednesday, June 8, 2005 by noon.
- Sign up for a final project review slot on Thu morning.

Announcements

- HW#5
 - ▷ Use 5-pt stencil discretization.
 - ▷ $f(x, y) = 1$.
 - ▷ For the timings, record the time for all iterations to convergence, not just the time for 1 iteration. Thus, you can compare the trade-off between parallelism and iteration count.
 - ▷ Note: as opposed to Jacobi, GS, and CG, different number of procs leads to *different* DDM methods.
 - ▷ It is advisable to first implement DDM using MATLAB, and then in parallel.

Final Project Guidelines

- Due date: Wed, June 8, noon.
- Submit a report which includes:
 - ▷ Description of the methodology/algorithm.
 - ▷ Description of the implementation.
 - ▷ Results (parallel efficiency, etc).
 - ▷ Conclusion, discussion, future work/what else needs to be done.
- Discussion: on 6/9, each student will meet one-on-one and discuss the project for about 20min. The idea is to help me better understand and evaluate your work. (See next page for more details.)
- Grading:
HW#1: 20%, HW#2: 20%, HW#3: 20%,
HW#4: 20%, HW#5/FP: 20%.

Overlapping Schwarz (2 Subdomains)

- Multiplicative (*Block Gauss-Seidel*)

$$\begin{cases} u^{k+1/2} \leftarrow u^k + \begin{pmatrix} A_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{pmatrix} (f - Au^k) \\ u^{k+1} \leftarrow u^{k+1/2} + \begin{pmatrix} 0 & 0 \\ 0 & A_{\Omega_2}^{-1} \end{pmatrix} (f - Au^{k+1/2}) \end{cases}$$

i.e.

$$u^{k+1} \leftarrow u^k + (B_1 + B_2 - B_2AB_1)(f - Au^k),$$

where $B_i =$ local correction w.r.t. Ω_i .

- Additive (*Block Jacobi*)

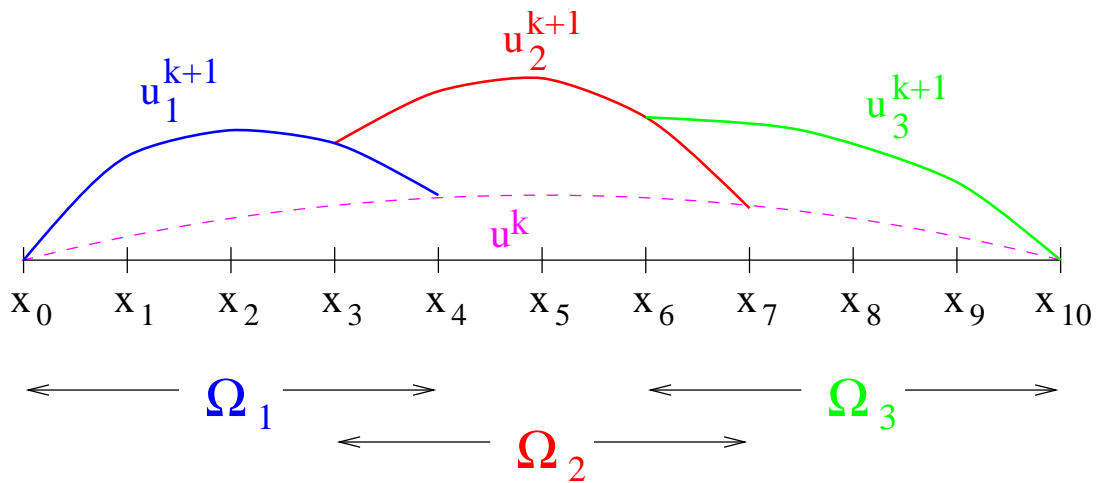
$$\begin{cases} u^{k+1/2} \leftarrow u^k + \begin{pmatrix} A_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{pmatrix} (f - Au^k) \\ u^{k+1} \leftarrow u^{k+1/2} + \begin{pmatrix} 0 & 0 \\ 0 & A_{\Omega_2}^{-1} \end{pmatrix} (f - Au^k) \end{cases}$$

i.e.

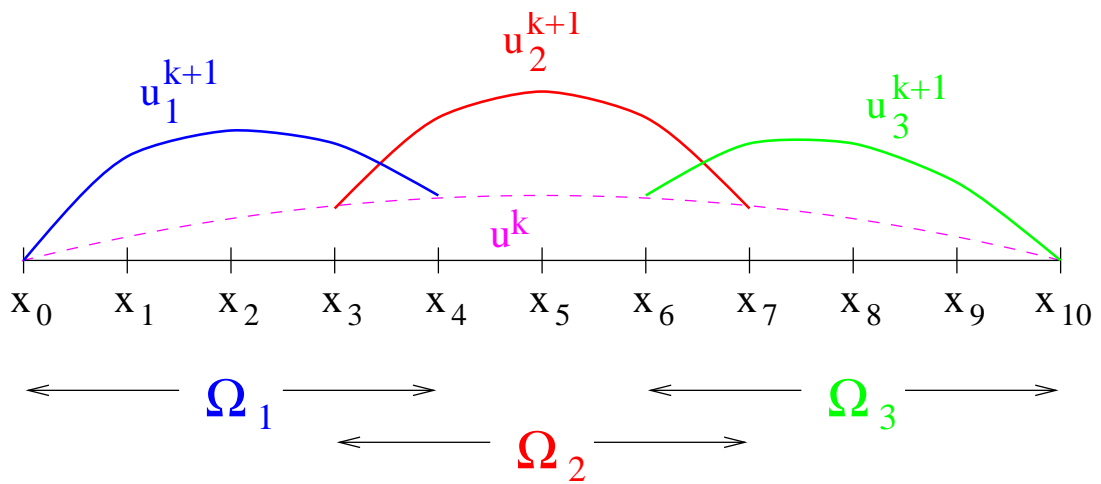
$$u^{k+1} \leftarrow u^k + (B_1 + B_2)(f - Au^k).$$

Example: Overlapping Schwarz

- Multiplicative



- Additive



Multiplicative Schwarz with Many Subdomains

- Domain decomposition methods are usually applied to many subdomains since:
 - ▷ Parallel computers usually have tens to thousands of processors
 - ▷ Easier to solve smaller subdomain problems.
- Suppose $\Omega = \cup_{i=1}^p \Omega_i$, a union of p overlapping subdomains. The multiplicative Schwarz method is:

$$\begin{aligned}u^{k+1/p} &\leftarrow u^k + B_1(f - Au^k) \\u^{k+2/p} &\leftarrow u^{k+1/p} + B_2(f - Au^{k+1/p}) \\&\vdots \\u^{k+1} &\leftarrow u^{k+(p-1)/p} + B_p(f - Au^{k+(p-1)/p})\end{aligned}$$

where $B_i = R_i^T A_i^{-1} R_i$, $A_i = R_i A R_i^T$.

- Symmetrized multiplicative Schwarz can be obtained by solving the subdomain problems from Ω_1 to Ω_p , and then from Ω_{p-1} to Ω_1 .

MS with Many Subdomains (cont.)

- The error $e^k = u - u^k$ satisfies:

$$e^{k+1} = (I - B_p A) \cdots (I - B_1 A) e^k$$

i.e. The preconditioner B_{MS} of MS with p subdomains is given by:

$$B_{MS} = [I - (I - B_p A) \cdots (I - B_1 A)] A^{-1}$$

- Like GS, MS is very sequential, but can be parallelized with a multicoloring technique. Subdomains with the same color can be solved in parallel.
- Suppose q colors are used. The multicolored MS iteration is:

$$u^{k+1/q} \leftarrow u^k + \sum_{i \in \text{Color}_1} B_i (f - Au^k)$$

$$u^{k+2/q} \leftarrow u^{k+1/q} + \sum_{i \in \text{Color}_2} B_i (f - Au^{k+1/q})$$

$$\begin{aligned} & \vdots \\ u^{k+1} & \leftarrow u^{k+(q-1)/q} + \sum_{i \in \text{Color}_q} B_i (f - Au^{k+(q-1)/q}) \end{aligned}$$

- In general, the convergence rate depends on q . Fewer colors typically means faster convergence.

Additive Schwarz with Many Subdomains

- Additive Schwarz iteration:

$$u^{k+1} \leftarrow u^k + \sum_{i=1}^p B_i(f - Au^k)$$

- The corresponding preconditioner is:

$$B_{AS} = \sum_{i=1}^p B_i$$

- Note: B_{AS} is symmetric if A is. Thus, no need to symmetrize.

Numerical Example (many subdomains)

PDE:

$$-\Delta u = xe^y \quad \text{in } \Omega$$

$$u = -xe^y \quad \text{on } \partial\Omega$$

Convergence results: iteration counts

n	Subdomains	MS with overlap size			AS with overlap size		
		1	2	4	1	2	4
33	2 × 1	3	3	2	5	5	4
33	4 × 1	4	4	3	7	6	5
33	8 × 1	8	6	4	9	8	6
33	2 × 2	4	3	3	7	6	4
33	ILU						10
33	SSOR						18
65	2 × 1	4	4	3	7	6	5
65	4 × 1	6	5	4	10	8	6
65	8 × 1	8	8	6	16	12	9
65	2 × 2	5	4	3	10	8	7
65	ILU						19
65	SSOR						62

$\Omega =$ unit square

Overlap	Subdomains	MS	AS
1	2	4	7
1	4	4	8
1	8	5	10
1	16	5	12
2	2	3	5
2	4	3	7
2	8	4	7
2	16	4	8
4	2	2	3
4	4	2	5
4	8	3	6
4	16	3	6
ILU	13		
SSOR	18		

$\Omega =$ unstructured grid

Observations:

- Suppose mesh size = $O(h)$, subdomain size = $O(H)$, and overlap size = $O(\delta)$.
- # of iters grows as $O(1/H)$.
- If $\delta \propto H$, # of iters is bounded indep. of h and H/h .
- # of iters for MS is about half of that for AS.
- Convergence is poor for $\delta = 0$, but improves rapidly as δ increases.

Nonoverlapping DD Methods

- Substructuring methods used in structural analysis.
- Complicated objects are usually built from connecting many individual parts together. It is convenient and sometimes advantageous to identify each substructure (or subdomain).
- Notations: subscript I denotes the portion of the matrix associated with nodes in the interior and on the true exterior boundary. Subscript B denotes those portions associated with the interior boundaries. Superscript (i) denotes subdomain number.

Direct Substructuring: 2 Substructures

- We partition the (stiffness) matrix A into:

$$A = \begin{pmatrix} A_{II}^{(1)} & 0 & A_{IB}^{(1)} \\ 0 & A_{II}^{(2)} & A_{IB}^{(2)} \\ A_{BI}^{(1)} & A_{BI}^{(2)} & A_{BB}^{(1)} + A_{BB}^{(2)} \end{pmatrix}$$

- $A_{II}^{(i)}$ = coupling between I nodes in Ω_i
 $A_{IB}^{(i)}$ = coupling between I & B nodes in Ω_i
 $A_{BB}^{(i)}$ = coupling between B nodes in Ω_i
- LDU factorization $\rightarrow A = LDU$:

$$L = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ A_{BI}^{(1)} A_{II}^{(1)-1} & A_{BI}^{(2)} A_{II}^{(2)-1} & I \end{pmatrix}$$

$$D = \begin{pmatrix} A_{II}^{(1)} & 0 & 0 \\ 0 & A_{II}^{(2)} & 0 \\ 0 & 0 & S^{(1)} + S^{(2)} \end{pmatrix}$$

$$U = \begin{pmatrix} I & 0 & A_{II}^{(1)-1} A_{IB}^{(1)} \\ 0 & I & A_{II}^{(2)-1} A_{IB}^{(2)} \\ 0 & 0 & I \end{pmatrix}$$

where

$$S^{(i)} = A_{BB}^{(i)} - A_{BI}^{(i)} A_{II}^{(i)-1} A_{IB}^{(i)},$$

the Schur complement w.r.t. subdomain Ω_i .

- Note: in general, $S^{(i)}$ are dense matrices.

- Partition $u = (u_I^{(1)}, u_I^{(2)}, u_B)^T$, and using the LDU factorization of A :

$$\begin{pmatrix} A_{II}^{(1)} & 0 & 0 \\ 0 & A_{II}^{(2)} & 0 \\ A_{BI}^{(1)} & A_{BI}^{(2)} & I \end{pmatrix} \begin{pmatrix} I & 0 & A_{II}^{(1)-1} A_{IB}^{(1)} \\ 0 & I & A_{II}^{(2)-1} A_{IB}^{(2)} \\ 0 & 0 & S^{(1)} + S^{(2)} \end{pmatrix} \begin{pmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_B \end{pmatrix} = \begin{pmatrix} f_I^{(1)} \\ f_I^{(2)} \\ f_B \end{pmatrix}$$

- By eliminating $u_I^{(1)}$ and $u_I^{(2)}$, we obtain the reduced Schur complement equation for u_B :

$$(S^{(1)} + S^{(2)})u_B = f_B - A_{BI}^{(1)} A_{II}^{(1)-1} f_I^{(1)} - A_{BI}^{(2)} A_{II}^{(2)-1} f_I^{(2)}$$

- $S^{(1)} + S^{(2)}$ is much smaller than the original matrix A , but also much denser.
- May repeat the same procedure to $S^{(1)} + S^{(2)} \rightarrow$ recursive LU factorization.
- For many subdomains. Let \tilde{R}_i be the restriction operator which returns the values on the interface of Ω_i :

$$u_B = \sum_{i=1}^p \tilde{R}_i^T u_B^{(i)}$$

- The Schur complement equation becomes:

$$\left(\sum_{i=1}^p \tilde{R}_i^T S^{(i)} \tilde{R}_i \right) u_B = f_B - \sum_{i=1}^p \tilde{R}_i^T A_{BI}^{(i)} A_{II}^{(i)-1} \tilde{R}_i f_I^{(i)}$$

Iterative Substructuring Methods: 2 Subdomains

- The action of the Schur complement $S^{(i)}$, where

$$S^{(i)} = A_{BB}^{(i)} - A_{BI}^{(i)} A_{II}^{(i)-1} A_{IB}^{(i)},$$

can be calculated with 3 matrix vector multiplies and 1 matrix solve.

- Thus, the linear system:

$$S u_B \equiv (S^{(1)} + S^{(2)}) u_B = g$$

can be solved iteratively by a (preconditioned) Krylov subsp method.

- Convergence rate of Krylov subsp methods depends on:

$$\kappa(S) = O\left(\frac{1}{h}\right) \quad \text{for PDE problems,}$$

compared to $\kappa(A) = O\left(\frac{1}{h^2}\right)$.

- Still need efficient preconditioner for S .

Neumann-Dirichlet Preconditioner

- Precondition $S = S^{(1)} + S^{(2)}$ by $S^{(1)^{-1}}$. The right preconditioned Schur complement equation becomes:

$$\begin{aligned} (S^{(1)} + S^{(2)})u_B &= g \\ (S^{(1)} + S^{(2)})S^{(1)^{-1}}w_B &= g \\ (I + S^{(2)}S^{(1)^{-1}})w_B &= g \end{aligned}$$

The resulting linear system is solved by a Krylov subsp method.

- How to compute $S^{(1)^{-1}}v$?

Note:

$$\begin{aligned} A^{(1)} &= \begin{pmatrix} I & 0 \\ A_{BI}^{(1)}A_{II}^{(1)^{-1}} & I \end{pmatrix} \begin{pmatrix} A_{II}^{(1)} & 0 \\ 0 & S^{(1)} \end{pmatrix} \begin{pmatrix} I & A_{II}^{(1)^{-1}}A_{IB}^{(1)} \\ 0 & I \end{pmatrix} \\ A^{(1)^{-1}} &= \begin{pmatrix} I & -A_{II}^{(1)^{-1}}A_{IB}^{(1)} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{II}^{(1)^{-1}} & 0 \\ 0 & S^{(1)^{-1}} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{BI}^{(1)}A_{II}^{(1)^{-1}} & I \end{pmatrix} \\ A^{(1)^{-1}} &= \begin{pmatrix} * & * \\ * & S^{(1)^{-1}} \end{pmatrix} \end{aligned}$$

$$\Rightarrow S^{(1)^{-1}}v = (0 \ I)A^{(1)^{-1}} \begin{pmatrix} 0 \\ I \end{pmatrix} v$$

- Hence, $S^{(1)^{-1}}v$ can be obtained by inverting $A^{(1)}$.

Neumann-Dirichlet Preconditioner (cont.)

- Note: $A^{(1)}$ is defined as:

$$A^{(1)} = \begin{pmatrix} A_{II}^{(1)} & A_{IB}^{(1)} \\ A_{BI}^{(1)} & A_{BB}^{(1)} \end{pmatrix}.$$

By definition of $A_{BB}^{(1)}$, it is not hard to see that inverting $A^{(1)}$ is equivalent to solving a PDE on Ω_1 with Neumann boundary condition on the interior interface.

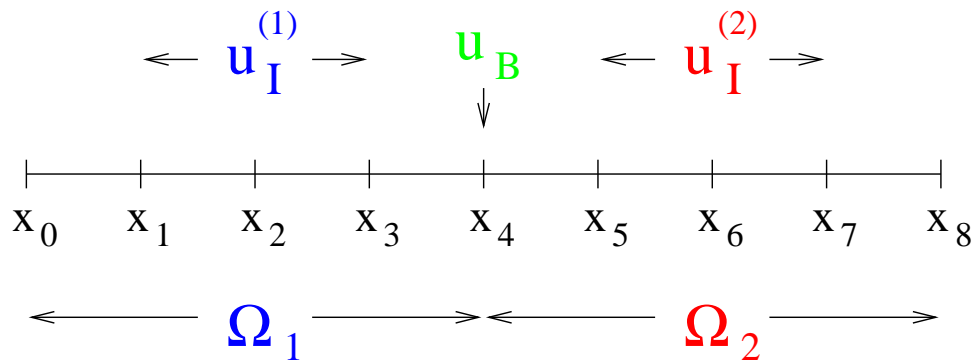
- To solve $(I + S^{(2)}S^{(1)^{-1}})$ by a Krylov subsp method, we also need to compute:

$$S^{(2)}v = (A_{BB}^{(2)} - A_{BI}^{(2)}A_{II}^{(2)^{-1}}A_{IB}^{(2)})v,$$

which requires inverting $A_{II}^{(2)}$. Similar to above, this is equivalent to solving a PDE on Ω_2 with Dirichlet boundary conditions on the artificial boundary $\partial\Omega_2$.

- Hence, this preconditioner is called Neumann-Dirichlet preconditioner.

Example: 1D, ND Preconditioner



- Neumann problem:

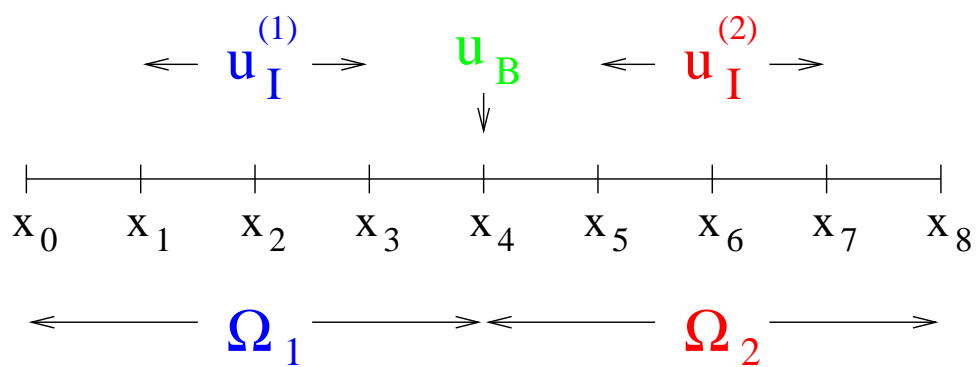
$$A^{(1)} = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 1 \end{pmatrix}$$

\Leftrightarrow

$$\begin{cases} \frac{d^2 u}{dx^2} = f & \text{in } (x_0, x_4) \\ u(x_0) = 0, \frac{du}{dx}(x_4) = 0 \end{cases}$$

- $S^{(1)}v = (0 \ 0 \ 0 \ 1) \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} v.$

Example: 1D, ND Preconditioner (cont.)



- Dirichlet problem:

$$\begin{aligned}
 S^{(2)} &= A_{BB}^{(2)} - A_{BI}^{(2)} A_{II}^{(2)-1} A_{IB}^{(2)} \\
 &= 1 - (-1 \ 0 \ 0) \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}^{-1} \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}.
 \end{aligned}$$

Neumann-Dirichlet Preconditioner (cont.)

The Neumann-Dirichlet preconditioner can be related directly to the original linear system.

- Solve for $v_I^{(i)}$ the 2 subdomain problems:

$$A_{II}^{(i)} v_I^{(i)} = f_I^{(i)}.$$

- Define $u_I^{(i)} = v_I^{(i)} + w_I^{(i)}$. Then $w_I^{(i)}$ satisfies:

$$\begin{pmatrix} A_{II}^{(1)} & A_{IB}^{(1)} & 0 \\ A_{BI}^{(1)} & A_{BB}^{(1)} + A_{BB}^{(2)} & A_{BI}^{(2)} \\ 0 & A_{IB}^{(2)} & A_{II}^{(2)} \end{pmatrix} \begin{pmatrix} w_I^{(1)} \\ u_B \\ w_I^{(2)} \end{pmatrix} = \begin{pmatrix} 0 \\ f_B \\ 0 \end{pmatrix}$$

- By direct computation:

$$\begin{aligned} & \begin{pmatrix} A_{II}^{(1)} & A_{IB}^{(1)} & 0 \\ A_{BI}^{(1)} & A_{BB}^{(1)} + A_{BB}^{(2)} & A_{BI}^{(2)} \\ 0 & A_{IB}^{(2)} & A_{II}^{(2)} \end{pmatrix} \begin{pmatrix} A_{II}^{(1)} & A_{IB}^{(1)} & 0 \\ A_{BI}^{(1)} & A_{BB}^{(1)} & 0 \\ 0 & A_{IB}^{(2)} & A_{II}^{(2)} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ y_B \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ (I + S^{(2)} S^{(1)-1}) y_B \\ 0 \end{pmatrix} \end{aligned}$$

Neumann-Dirichlet Preconditioner (cont.)

- Thus, applying ND preconditioner to a vector y restricted to the interior boundary corresponding to solving the block triangular system:

$$\begin{pmatrix} A_{II}^{(1)} & A_{IB}^{(1)} & 0 \\ A_{BI}^{(1)} & A_{BB}^{(1)} & 0 \\ 0 & A_{IB}^{(2)} & A_{II}^{(2)} \end{pmatrix}^{-1}$$

which requires:

$$\begin{pmatrix} A_{II}^{(1)} & A_{IB}^{(1)} \\ A_{BI}^{(1)} & A_{BB}^{(1)} \end{pmatrix}^{-1} \Leftrightarrow \text{Neumann problem}$$

and

$$A_{II}^{(2)-1} \Leftrightarrow \text{Dirichlet problem}$$

Dirichlet-Neumann Preconditioner

- Precondition the original linear system from the left using a similar upper block triangular system:

$$\begin{aligned}
 & \begin{pmatrix} A_{II}^{(1)} & A_{IB}^{(1)} & 0 \\ A_{BI}^{(1)} & A_{BB}^{(1)} & A_{BI}^{(2)} \\ 0 & 0 & A_{II}^{(2)} \end{pmatrix}^{-1} \begin{pmatrix} A_{II}^{(1)} & A_{IB}^{(1)} & 0 \\ A_{BI}^{(1)} & A_{BB}^{(1)} + A_{BB}^{(2)} & A_{BI}^{(2)} \\ 0 & A_{IB}^{(2)} & A_{II}^{(2)} \end{pmatrix} \begin{pmatrix} 0 \\ y_B \\ 0 \end{pmatrix} \\
 &= \begin{pmatrix} * \\ (I + S^{(1)-1} S^{(2)}) y_B \\ * \end{pmatrix}
 \end{aligned}$$

- A Dirichlet problem ($A_{II}^{(2)-1}$) is solved first, followed by a Neumann problem ($A^{(1)-1}$).

Neumann-Neumann Preconditioner

- $S^{-1} \approx S^{(1)-1} + S^{(2)-1}$, i.e.

$$(S^{(1)} + S^{(2)})(S^{(1)-1} + S^{(2)-1})w_B = g$$

- Two Neumann problems $(S^{(1)-1}, S^{(2)-1})$ and two Dirichlet problems $(S^{(1)}, S^{(2)})$ are solved.
- Convergence independent of h , but depends on PDE and sizes of subdomains H .
- For 2 subdomains, NN preconditioner has no advantages over ND preconditioner. However, it has more general applicability for many subdomains.

Many Subdomains

Neumann-Dirichlet preconditioner:

- Write the Schur complement as

$$S = \sum_{\Omega_i} \tilde{R}_i^T S_i \tilde{R}_i$$

- If one can make a red-black coloring of the subdomains $\{\Omega_i\}$, rewrite S as:

$$S = \sum_{\Omega_i^R} \tilde{R}_i^{(R)T} S_i^{(R)} \tilde{R}_i^{(R)} + \sum_{\Omega_i^B} \tilde{R}_i^{(B)T} S_i^{(B)} \tilde{R}_i^{(B)}$$

- The ND preconditioner B_{ND} is given by:

$$B_{ND} = \sum_{\Omega_i^R} \tilde{R}_i^{(R)T} S_i^{(R)-1} \tilde{R}_i^{(R)}$$

and SB_{ND} is:

$$\sum_{\Omega_i^R} \tilde{R}_i^{(R)T} \tilde{R}_i^{(R)} + \left(\sum_{\Omega_i^B} \tilde{R}_i^{(B)T} S_i^{(B)} \tilde{R}_i^{(B)} \right) \left(\sum_{\Omega_i^R} \tilde{R}_i^{(R)T} S_i^{(R)-1} \tilde{R}_i^{(R)} \right)$$

- Note: first term = identity matrix when there are no crosspoints shared by subdomains having the same color.

Many Subdomains (cont.)

Neumann-Neumann preconditioner:

- Define a diagonal (scaling) matrix D such that for each element of u_B , D_{jj}^{-1} is the number of subdomains which share the node j .
- The NN preconditioner B_{NN} is defined as:

$$B_{NN} = D \left(\sum_{\Omega_i} \tilde{R}_i^T S^{(i)-1} \tilde{R}_i \right) D$$

or more generally,

$$B_{NN} = \sum_{\Omega_i} D_i \tilde{R}_i^T S^{(i)-1} \tilde{R}_i D_i.$$