

Lecture : Domain decomposition



School of Mathematical Sciences, Xiamen University

1. Motivations for domain decomposition

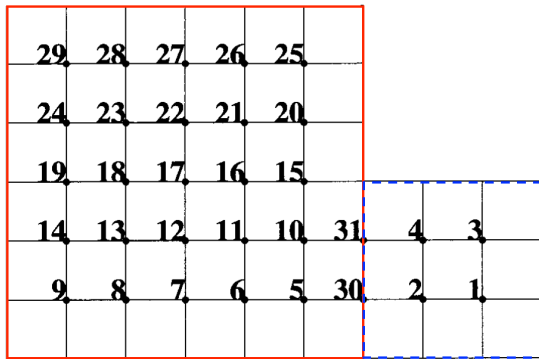
- The region of solution of real problems is irregular. Even more complicated problems such as that with different equations in different regions.
- The problem is too large to fit in the computer memory and may have to be solved “in pieces”.
- We may want to break the problem into subproblems that can be solved in parallel on a parallel computer.

2. The idea of domain decomposition

- Domain decomposition breaks a large problem into subproblems, then uses simpler methods solve the individual subproblems, and finally combines the solutions together to get the overall solution.
- These subproblems can be solved one at a time if the whole problem does not fit into memory, or in parallel on a parallel computer.

3. Nonoverlapping methods

- This method is also called *substructuring* or a *Schur complement method* in the literature.
- Example: Poisson's equation with Dirichlet boundary conditions on an L-shaped region discretized with a 5-point stencil.



- The discretized matrix

4 -1 -1	-1 4 -1						-1
-1 4 -1	-1 -1 4						-1
		4 -1 -1 -1 4 -1 -1 4 -1 -1 4 -1 -1 4	-1 -1 -1 -1				-1
		-1 -1 -1 -1 -1 -1 -1 -1 -1	4 -1 -1 -1 4 -1 -1 4 -1 -1 4 -1 -1 4	-1 -1 -1 -1 -1			-1
			-1 -1 -1 -1 -1 -1 -1	4 -1 -1 -1 4 -1 -1 4 -1 -1 4 -1 -1 4	-1 -1 -1 -1 -1		
				-1 -1 -1 -1 -1 -1 -1	4 -1 -1 -1 4 -1 -1 4 -1 -1 4 -1 -1 4	-1 -1 -1 -1 -1	
					-1 -1 -1 -1 -1 -1 -1	4 -1 -1 -1 4 -1 -1 4 -1 -1 4 -1 -1 4	-1 -1 -1 -1 -1
						-1 -1 -1 -1 -1 -1 -1	4 -1 -1 -1 4 -1 -1 4 -1 -1 4 -1 -1 4
-1	-1	-1	-1				4 -1 -1 4

- The discretized matrix is a block 3×3 matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} & \mathbf{A}_{13} \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{13}^\top & \mathbf{A}_{23}^\top & \mathbf{A}_{33} \end{bmatrix}.$$

Note that $\mathbf{A}_{12} = \mathbf{0}$, since there is no direct coupling between the interior grid points of the two subdomains.

- The block LDU decomposition of \mathbf{A} is

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{13}^\top \mathbf{A}_{11}^{-1} & \mathbf{A}_{23}^\top \mathbf{A}_{22}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} & \mathbf{A}_{13} \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix},$$

where

$$\mathbf{S} = \mathbf{A}_{33} - \mathbf{A}_{13}^\top \mathbf{A}_{11}^{-1} \mathbf{A}_{13} - \mathbf{A}_{23}^\top \mathbf{A}_{22}^{-1} \mathbf{A}_{23}$$

is called the Schur complement of the leading principal submatrix containing \mathbf{A}_{11} and \mathbf{A}_{22} .

- Then we have $\mathbf{A}^{-1} =$

$$\begin{bmatrix} \mathbf{A}_{11}^{-1} & \mathbf{0} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{13} \\ \mathbf{0} & \mathbf{A}_{22}^{-1} & -\mathbf{A}_{22}^{-1}\mathbf{A}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{A}_{13}^{\top}\mathbf{A}_{11}^{-1} & -\mathbf{A}_{23}^{\top}\mathbf{A}_{22}^{-1} & \mathbf{I} \end{bmatrix}.$$

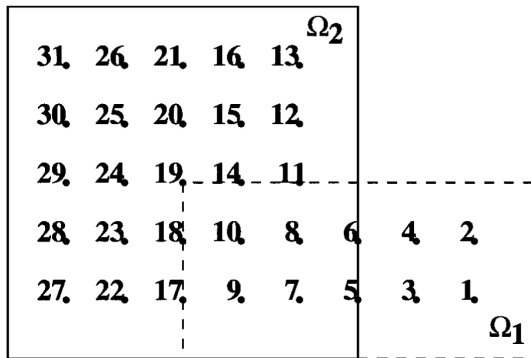
To multiply a vector by \mathbf{A}^{-1} we need to multiply by the blocks in the entries of this factored form of \mathbf{A}^{-1} , namely, \mathbf{A}_{13} and \mathbf{A}_{23} (and their transposes), \mathbf{A}_{11}^{-1} , \mathbf{A}_{22}^{-1} , and \mathbf{S}^{-1} . Multiplying by \mathbf{A}_{13} and \mathbf{A}_{23} is cheap, and multiplying by \mathbf{A}_{11}^{-1} and \mathbf{A}_{22}^{-1} is also cheap. It remains to explain how to multiply by \mathbf{S}^{-1} . Preconditioned Krylov iterative methods are preferred.

- The case for k subdomains is a block $(k+1) \times (k+1)$ matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{0} & \mathbf{A}_{1,k+1} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{A}_{kk} & \mathbf{A}_{k,k+1} \\ \mathbf{A}_{1,k+1}^{\top} & \cdots & \mathbf{A}_{k,k+1}^{\top} & \mathbf{A}_{k+1,k+1} \end{bmatrix}.$$

4. Overlapping methods (Schwarz domain decomposition)

- In nonoverlapping methods, the domains corresponding to the nodes in \mathbf{A}_{ii} were disjoint. In overlapping methods, we use overlapping domains.
- Example: Poisson's equation with Dirichlet boundary conditions on an L-shaped region discretized with a 5-point stencil.



- The discretized matrix

[illegible]

- Partition \mathbf{A} as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\Omega_1, \Omega_1} & \mathbf{A}_{\Omega_1, \Omega \setminus \Omega_1} \\ \mathbf{A}_{\Omega \setminus \Omega_1, \Omega_1} & \mathbf{A}_{\Omega \setminus \Omega_1, \Omega \setminus \Omega_1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\Omega \setminus \Omega_2, \Omega \setminus \Omega_2} & \mathbf{A}_{\Omega \setminus \Omega_2, \Omega_2} \\ \mathbf{A}_{\Omega_2, \Omega \setminus \Omega_2} & \mathbf{A}_{\Omega_2, \Omega_2} \end{bmatrix}.$$

- Partition \mathbf{x} conformally:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{\Omega_1} \\ \mathbf{x}_{\Omega \setminus \Omega_1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{1:10} \\ \mathbf{x}_{11:31} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{\Omega \setminus \Omega_2} \\ \mathbf{x}_{\Omega_2} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{1:6} \\ \mathbf{x}_{7:31} \end{bmatrix}.$$

4.1. Additive Schwarz

Algorithm: Additive Schwarz or overlapping block Jacobi

$$\begin{aligned} \mathbf{r} &= \mathbf{b} - \mathbf{A}\mathbf{x}^{(i)}; && \text{compute residual} \\ \mathbf{x}^{(i+1)} &= \mathbf{x}^{(i)}; \\ \mathbf{x}_{\Omega_1}^{(i+1)} &= \mathbf{x}_{\Omega_1}^{(i)} + \mathbf{A}_{\Omega_1, \Omega_1}^{-1} \mathbf{r}_{\Omega_1}; && \text{update solution on } \Omega_1 \\ \mathbf{x}_{\Omega_2}^{(i+1)} &= \mathbf{x}_{\Omega_2}^{(i+1)} + \mathbf{A}_{\Omega_2, \Omega_2}^{-1} \mathbf{r}_{\Omega_2}; && \text{update solution on } \Omega_2 \end{aligned}$$

- This algorithm also can be written in one line as

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \begin{bmatrix} \mathbf{A}_{\Omega_1, \Omega_1}^{-1} \mathbf{r}_{\Omega_1} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_{\Omega_2, \Omega_2}^{-1} \mathbf{r}_{\Omega_2} \end{bmatrix}.$$

- Since the additive Schwarz method is iterative, it is not necessary to solve the problems on Ω_i exactly, i.e., $\mathbf{A}_{\Omega_1, \Omega_1}^{-1} \mathbf{r}_{\Omega_1}$ and $\mathbf{A}_{\Omega_2, \Omega_2}^{-1} \mathbf{r}_{\Omega_2}$ can be computed inexactly.
- Indeed, the additive Schwarz method is typically used as a preconditioner for a Krylov subspace method. The preconditioner \mathbf{M} is given by

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A}_{\Omega_1, \Omega_1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\Omega_2, \Omega_2}^{-1} \end{bmatrix}.$$

If Ω_1 and Ω_2 do not overlap, then $\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A}_{\Omega_1, \Omega_1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\Omega_2, \Omega_2}^{-1} \end{bmatrix}.$

4.2. Multiplicative Schwarz

Algorithm: Multiplicative Schwarz or overlapping block G–S

$$\mathbf{r}_{\Omega_1} = (\mathbf{b} - \mathbf{A}\mathbf{x}^{(i)})_{\Omega_1}; \quad \text{compute residual on } \Omega_1$$

$$\mathbf{x}^{(i+1/2)} = \mathbf{x}^{(i)};$$

$$\mathbf{x}_{\Omega_1}^{(i+1/2)} = \mathbf{x}_{\Omega_1}^{(i)} + \mathbf{A}_{\Omega_1, \Omega_1}^{-1} \mathbf{r}_{\Omega_1}; \quad \text{update solution on } \Omega_1$$

$$\mathbf{r}_{\Omega_2} = (\mathbf{b} - \mathbf{A}\mathbf{x}^{(i+1/2)})_{\Omega_2}; \quad \text{compute residual on } \Omega_2$$

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i+1/2)};$$

$$\mathbf{x}_{\Omega_2}^{(i+1)} = \mathbf{x}_{\Omega_2}^{(i+1/2)} + \mathbf{A}_{\Omega_2, \Omega_2}^{-1} \mathbf{r}_{\Omega_2}; \quad \text{update solution on } \Omega_2$$

- This algorithm first solves Poisson's equation on Ω_1 using boundary data from $\mathbf{x}^{(i)}$. It then solves Poisson's equation on Ω_2 , but using boundary data that has just been updated.

- **Exercise:** The multiplicative Schwarz method also can be used as a preconditioner for a Krylov subspace method. What is the corresponding preconditioner?
- In practice more domains than just two (Ω_1 and Ω_2) are used. This is done if the domain of solution is more complicated or if there are many independent parallel processors available to solve independent problems $\mathbf{A}_{\Omega_i, \Omega_i}^{-1} \mathbf{r}_{\Omega_i}$, or just to keep the subproblems $\mathbf{A}_{\Omega_i, \Omega_i}^{-1} \mathbf{r}_{\Omega_i}$ small and inexpensive to solve.
- Two-level correction: $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i+1)} + \mathbf{R}^\top \mathbf{A}_H^{-1} \mathbf{R}(\mathbf{b} - \mathbf{A} \mathbf{x}^{(i+1)})$

5. Further reading

- **Iterative Methods and Preconditioners for Systems of Linear Equations**

Gabriele Ciaramella and Martin J. Gander

SIAM, 2022