Iterative methods for linear algebra

Pierre Jolivet `pierre@joliv.et`
Ronan Guivarch `ronan.guivarch@toulouse-inp.fr`

2/2 *Advanced preconditioning*

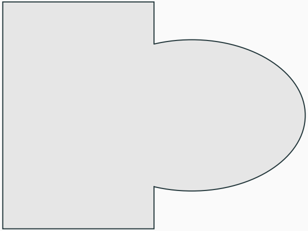
Table of contents

1. Introduction
2. Overlapping Schwarz methods
3. Coarse space operators
4. Substructuring methods
5. Implementation aspects
6. Links with multigrid methods

Introduction

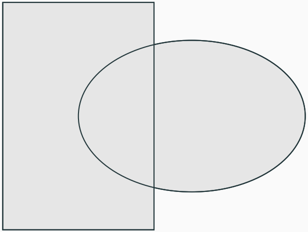
Motivation

How to solve Poisson equation on “complex” domains?



Motivation

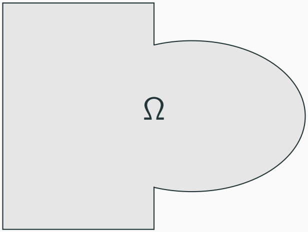
How to solve Poisson equation on “complex” domains?



“Simpler” domains and Fourier transforms [Schwarz 1870]

Overlapping Schwarz methods

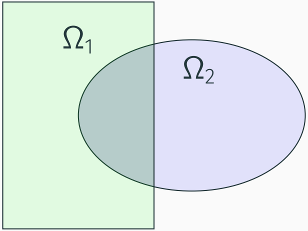
Continuous formulation



Given $\mathcal{L}$, a source term f, and Dirichlet BC g,

$$\begin{aligned} \mathcal{L}u &= f \text{ in } \Omega \\ u &= g \text{ on } \partial\Omega \end{aligned}$$

Continuous formulation



Given $\mathcal{L}$, a source term f, and Dirichlet BC g + $u_2^{(0)}$,

$$
\begin{aligned}
\mathcal{L}u_1^{(n+1)} &= f|_{\Omega_1} && \text{in } \Omega_1 & \qquad \mathcal{L}u_2^{(n+1)} &= f|_{\Omega_2} && \text{in } \Omega_2 \\
u_1^{(n+1)} &= g && \text{on } \partial\Omega_1 \setminus \Omega_2 & \qquad u_2^{(n+1)} &= g && \text{on } \partial\Omega_2 \setminus \Omega_1 \\
u_1^{(n+1)} &= u_2^{(n)} && \text{on } \partial\Omega_1 \cap \Omega_2 & \qquad u_2^{(n+1)} &= u_1^{(n+1)} && \text{on } \partial\Omega_2 \cap \Omega_1
\end{aligned}
$$

Error estimates

The errors $\{e_i^{(n+1)} = u_i^{(n+1)} - u|_{\Omega_i}\}_{i\in[\![1;2]\!]}$ verify:

$$\begin{aligned} \mathcal{L}e_1^{(n+1)} &= 0|_{\Omega_1} \text{ in } \Omega_1 & \mathcal{L}e_2^{(n+1)} &= 0|_{\Omega_2} \text{ in } \Omega_2 \\ e_1^{(n+1)} &= 0 \quad \text{on } \partial\Omega_1 \setminus \Omega_2 & e_2^{(n+1)} &= 0 \quad \text{on } \partial\Omega_2 \setminus \Omega_1 \\ e_1^{(n+1)} &= e_2^{(n)} \text{ on } \partial\Omega_1 \cap \Omega_2 & e_2^{(n+1)} &= e_1^{(n+1)} \text{ on } \partial\Omega_2 \cap \Omega_1 \end{aligned}$$

Error estimates

The errors $\{e_i^{(n+1)} = u_i^{(n+1)} - u|_{\Omega_i}\}_{i\in[\![1;2]\!]}$ verify:

$$\begin{aligned} \mathcal{L}e_1^{(n+1)} &= 0|_{\Omega_1} && \text{in } \Omega_1 \\ e_1^{(n+1)} &= 0 && \text{on } \partial\Omega_1 \setminus \Omega_2 \\ e_1^{(n+1)} &= e_2^{(n)} && \text{on } \partial\Omega_1 \cap \Omega_2 \end{aligned} \qquad \begin{aligned} \mathcal{L}e_2^{(n+1)} &= 0|_{\Omega_2} && \text{in } \Omega_2 \\ e_2^{(n+1)} &= 0 && \text{on } \partial\Omega_2 \setminus \Omega_1 \\ e_2^{(n+1)} &= e_1^{(n+1)} && \text{on } \partial\Omega_2 \cap \Omega_1 \end{aligned}$$

Convergence rates

- 1D, $\mathcal{L} = -\dfrac{\partial^2}{\partial x^2} \implies \rho(\delta) = \left|\dfrac{e_2^{(n+1)}(L_1)}{e_2^{(n)}(L_1)}\right| = \dfrac{1-\delta/(L-l_2)}{1+\delta/l_2}$
- 2D, $\mathcal{L} = (\nu - \Delta) \implies \rho(k,\delta,\nu) = \left|\dfrac{\hat{e}_1^{(n+1)}(0)}{\hat{e}_1^{(n)}(0)}\right| = e^{-2\delta\sqrt{\nu+k^2}}$

Remarks about the convergence rates

- both methods converge $\iff \delta > 0$

Remarks about the convergence rates

- both methods converge $\iff \delta > 0$
- for $k_1 > k_2$, $\rho(k_2, \delta, \nu) \gg \rho(k_1, \delta, \nu)$ (smoothing property)

Given $\left(u_1^{(0)}, u_2^{(0)}\right)$, [Lions 1988]:

$$
\begin{aligned}
\mathcal{L}u_1^{(n+1)} &= f|_{\Omega_1} \text{ in } \Omega_1 & \qquad \mathcal{L}u_2^{(n+1)} &= f|_{\Omega_2} \text{ in } \Omega_2 \\
u_1^{(n+1)} &= g \quad \text{ on } \partial\Omega_1 \setminus \Omega_2 & u_2^{(n+1)} &= g \quad \text{ on } \partial\Omega_2 \setminus \Omega_1 \\
u_1^{(n+1)} &= u_2^{(n)} \text{ on } \partial\Omega_1 \cap \Omega_2 & u_2^{(n+1)} &= u_1^{(n)} \text{ on } \partial\Omega_2 \cap \Omega_1
\end{aligned}
$$

// variant of the alternating Schwarz method

Given $\left(u_1^{(0)}, u_2^{(0)}\right)$, [Lions 1988]:

$$\begin{aligned} \mathcal{L}u_1^{(n+1)} &= f|_{\Omega_1} \text{ in } \Omega_1 \\ u_1^{(n+1)} &= g \quad \text{ on } \partial\Omega_1 \setminus \Omega_2 \\ u_1^{(n+1)} &= u_2^{(n)} \text{ on } \partial\Omega_1 \cap \Omega_2 \end{aligned} \qquad \begin{aligned} \mathcal{L}u_2^{(n+1)} &= f|_{\Omega_2} \text{ in } \Omega_2 \\ u_2^{(n+1)} &= g \quad \text{ on } \partial\Omega_2 \setminus \Omega_1 \\ u_2^{(n+1)} &= u_1^{(n)} \text{ on } \partial\Omega_2 \cap \Omega_1 \end{aligned}$$

Slower convergence but more suited to parallel computing

Connection with the block Gauss–Seidel method

Let $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and $M = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}$

given x_0

$$x_{k+1} = x_k + M^{-1} r_k$$

Connection with the block Gauss–Seidel method

Let $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and $M = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}$

given x_0

$$x_{k+1} = x_k + M^{-1} r_k$$

In order to explicit the workload distribution:

given x_0

$$x_{k+1/2} = x_k + \left(R_1^T (R_1 A R_1^T)^{-1} R_1\right) r_k$$

$$x_{k+1} = x_{k+1/2} + \left(R_2^T (R_2 A R_2^T)^{-1} R_2\right) r_{k+1/2}$$

with appropriate restriction operators $\{R_i\}_{i \in [\![1;2]\!]}$

Iteration matrix/local projections

Let $\{B_i = R_i^T(R_iAR_i^T)^{-1}R_i\}_{i\in[\![1;2]\!]}$

$$x_{k+1} = x_k + (B_1 + B_2 - B_2AB_1)r_k$$

Iteration matrix/local projections

Let $\{B_i = R_i^T(R_iAR_i^T)^{-1}R_i\}_{i\in[\![1;2]\!]}$

$$\begin{aligned} x_{k+1} &= x_k + (B_1 + B_2 - B_2AB_1)r_k \\ e_{k+1} &= (I - M^{-1}A)e_k \\ &= (I - B_2A)(I - B_1A)e_k \end{aligned}$$

A-orthogonal projection on span(R_i^T): $P_i = B_iA$

Iteration matrix/local projections

Let $\{B_i = R_i^T (R_i A R_i^T)^{-1} R_i\}_{i \in [\![1;2]\!]}$

$$
\begin{aligned}
x_{k+1} &= x_k + (B_1 + B_2 - B_2 A B_1) r_k \\
e_{k+1} &= (I - M^{-1} A) e_k \\
&= (I - B_2 A)(I - B_1 A) e_k
\end{aligned}
$$

A-orthogonal projection on span(R_i^T): $P_i = B_i A$

Moreover, $B_i r = P_i e$

Iteration matrix/local projections

Let $\{B_i = R_i^T(R_i A R_i^T)^{-1} R_i\}_{i \in [\![1;2]\!]}$

$$\begin{aligned} x_{k+1} &= x_k + (B_1 + B_2 - B_2 A B_1) r_k \\ e_{k+1} &= (I - M^{-1}A) e_k \\ &= (I - B_2 A)(I - B_1 A) e_k \end{aligned}$$

A-orthogonal projection on $\mathrm{span}(R_i^T)$: $P_i = B_i A$

Moreover, $B_i r = P_i e$

A-orthogonal projection on $\mathrm{span}(R_i^T)^{\perp_A}$: $I - P_i$

Connection with the block Jacobi method

Let $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and $M = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$

given x_0

$$x_{k+1} = x_k + M^{-1} r_k$$

Connection with the block Jacobi method

Let $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and $M = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$

given x_0

$$x_{k+1} = x_k + M^{-1} r_k$$

In order to explicit the parallelism:

given x_0

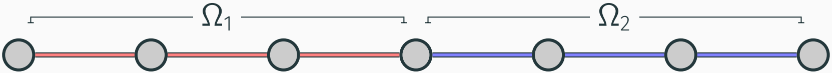
$$x_{k+1} = x_k + \left(R_1^T (R_1 A R_1^T)^{-1} R_1 + R_2^T (R_2 A R_2^T)^{-1} R_2 \right) r_k$$

1D domain

Algebraic overlap

1D domain

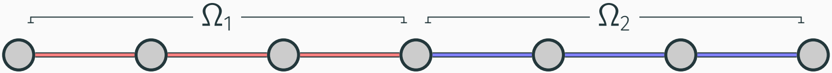
Initial decomposition



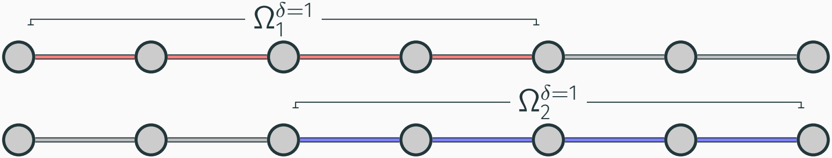
Algebraic overlap

1D domain

Initial decomposition



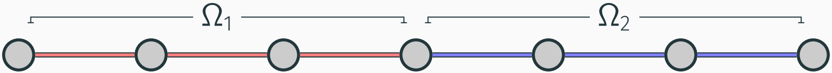
Overlapping decomposition



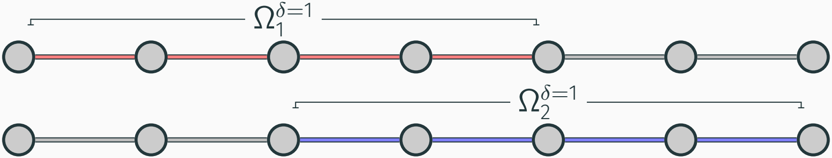
Algebraic overlap

1D domain

Initial decomposition



Overlapping decomposition



Larger restriction operators $\{R_i^\delta\}_{i \in [\![1;2]\!]}$

Alternating Schwarz method

Just as for the block GS method, with N subdomains:

given x_0

$$x_{k+1/N} = x_k + \left(R_1^{\delta\,T}(R_1^\delta A R_1^{\delta\,T})^{-1} R_1^\delta\right) r_k$$

$$\vdots$$

$$x_{k+1} = x_{k+(N-1)/N} + \left(R_N^{\delta\,T}(R_N^\delta A R_N^{\delta\,T})^{-1} R_N^\delta\right) r_{k+(N-1)/N}$$

Alternating Schwarz method

Just as for the block GS method, with N subdomains:

given x_0

$$x_{k+1/N} = x_k + \left({R_1^\delta}^T (R_1^\delta A {R_1^\delta}^T)^{-1} R_1^\delta \right) r_k$$

$$\vdots$$

$$x_{k+1} = x_{k+(N-1)/N} + \left({R_N^\delta}^T (R_N^\delta A {R_N^\delta}^T)^{-1} R_N^\delta \right) r_{k+(N-1)/N}$$

Red–Black coloring of subdomains for better concurrency

Additive Schwarz method

Just as for the block Jacobi method, with N subdomains:

given x_0

$$
\begin{aligned}
x_{k+1} &= x_k \\
x_{k+1} &\mathrel{+}= \left({R_1^\delta}^T (R_1^\delta A {R_1^\delta}^T)^{-1} R_1^\delta \right) r_k \\
&\vdots \\
x_{k+1} &\mathrel{+}= \left({R_N^\delta}^T (R_N^\delta A {R_N^\delta}^T)^{-1} R_N^\delta \right) r_k
\end{aligned}
$$

Additive Schwarz method

Just as for the block Jacobi method, with N subdomains:

$$
\begin{aligned}
&\text{given } x_0 \\
&x_{k+1} = x_k \\
&x_{k+1} \mathrel{+}= \left({R_1^\delta}^T (R_1^\delta A {R_1^\delta}^T)^{-1} R_1^\delta \right) r_k \\
&\quad\vdots \\
&x_{k+1} \mathrel{+}= \left({R_N^\delta}^T (R_N^\delta A {R_N^\delta}^T)^{-1} R_N^\delta \right) r_k
\end{aligned}
$$

This fixed-point algorithm does not converge

The error in the overlap does not decrease

Restricted additive Schwarz method

Partition of unity

A set of diagonal matrices $\{D_i\}_{i \in [\![1;N]\!]}$:

$$\sum_{i=1}^{N} R_i^T D_i R_i = I$$

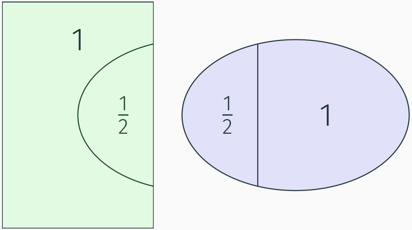
Restricted additive Schwarz method

Partition of unity

A set of diagonal matrices $\{D_i\}_{i\in[\![1;N]\!]}$:

$$\sum_{i=1}^{N} R_i^T D_i R_i = I$$

Multiplicity scaling



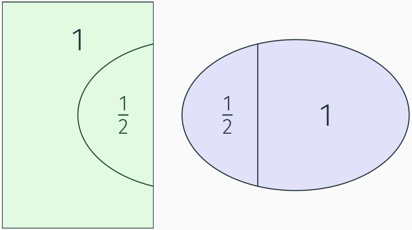
Restricted additive Schwarz method

Partition of unity

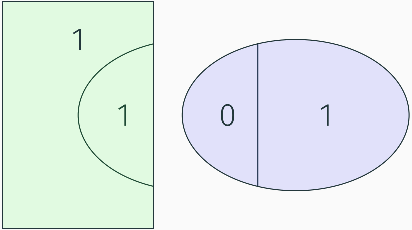
A set of diagonal matrices $\{D_i\}_{i\in[\![1;N]\!]}$:

$$\sum_{i=1}^{N} R_i^T D_i R_i = I$$

Multiplicity scaling



Master–slave



Restricted additive Schwarz method

Partition of unity

A set of diagonal matrices $\{D_i\}_{i \in [\![1;N]\!]}$:

$$\sum_{i=1}^{N} R_i^T D_i R_i = I$$

$$\begin{aligned}
&\text{given } x_0 \\
&\qquad x_{k+1} = x_k \\
&\qquad x_{k+1} \mathrel{+}= \left(R_1^T D_1 (R_1 A R_1^T)^{-1} R_1\right) r_k \\
&\qquad\qquad \vdots \\
&\qquad x_{k+1} \mathrel{+}= \left(R_N^T D_N (R_N A R_N^T)^{-1} R_N\right) r_k
\end{aligned}$$

Introduced by [Cai and Sarkis 1999]

Schwarz methods as preconditioners

Fixed-point algorithms as preconditioners, e.g.:

$$M_{\text{ASM}}^{-1} = \sum_{i=1}^{N} R_i^T (R_i A R_i^T)^{-1} R_i$$

$$M_{\text{RAS}}^{-1} = \sum_{i=1}^{N} R_i^T D_i (R_i A R_i^T)^{-1} R_i$$

Schwarz methods as preconditioners

Fixed-point algorithms as preconditioners, e.g.:

$$M_{\text{ASM}}^{-1} = \sum_{i=1}^{N} R_i^T (R_i A R_i^T)^{-1} R_i$$

$$M_{\text{RAS}}^{-1} = \sum_{i=1}^{N} R_i^T D_i (R_i A R_i^T)^{-1} R_i$$

Better convergence inside a Krylov method

M_{ASM}^{-1} does actually converge

Optimized boundary conditions

Instead of considering Dirichlet BC, use Robin BC

Let $\left\{\mathcal{B}_i = \frac{\partial}{\partial n_i} + \alpha\right\}_{i\in[\![1;2]\!]}$, for $\alpha \in \mathbb{R}^{+\star}$

$$
\begin{aligned}
\mathcal{L}u_1^{(n+1)} &= f|_{\Omega_1} && \text{in } \Omega_1 & \qquad \mathcal{L}u_2^{(n+1)} &= f|_{\Omega_2} && \text{in } \Omega_2 \\
u_1^{(n+1)} &= g && \text{on } \partial\Omega_1 \setminus \Omega_2 & \qquad u_2^{(n+1)} &= g && \text{on } \partial\Omega_2 \setminus \Omega_1 \\
\mathcal{B}_1 u_1^{(n+1)} &= \mathcal{B}_1 u_2^{(n)} && \text{on } \partial\Omega_1 \cap \Omega_2 & \qquad \mathcal{B}_2 u_2^{(n+1)} &= \mathcal{B}_2 u_1^{(n+1)} && \text{on } \partial\Omega_2 \cap \Omega_1
\end{aligned}
$$

Optimized boundary conditions

Instead of considering Dirichlet BC, use Robin BC

Let $\left\{\mathcal{B}_i = \frac{\partial}{\partial n_i} + \alpha\right\}_{i\in[\![1;2]\!]}$, for $\alpha \in \mathbb{R}^{+\star}$

$$\begin{aligned}
\mathcal{L}u_1^{(n+1)} &= f|_{\Omega_1} && \text{in } \Omega_1 \\
u_1^{(n+1)} &= g && \text{on } \partial\Omega_1 \setminus \Omega_2 \\
\mathcal{B}_1 u_1^{(n+1)} &= \mathcal{B}_1 u_2^{(n)} && \text{on } \partial\Omega_1 \cap \Omega_2
\end{aligned}
\qquad
\begin{aligned}
\mathcal{L}u_2^{(n+1)} &= f|_{\Omega_2} && \text{in } \Omega_2 \\
u_2^{(n+1)} &= g && \text{on } \partial\Omega_2 \setminus \Omega_1 \\
\mathcal{B}_2 u_2^{(n+1)} &= \mathcal{B}_2 u_1^{(n+1)} && \text{on } \partial\Omega_2 \cap \Omega_1
\end{aligned}$$

Convergence rate

- $\mathcal{L} = (\nu - \Delta) \implies \rho(k, \delta, \nu, \alpha) = \left|\frac{\sqrt{\nu + k^2} - \alpha}{\sqrt{\nu + k^2} + \alpha}\right| e^{-\delta\sqrt{\nu + k^2}}$

Optimal boundary conditions

Given general BC operators $\{\mathcal{B}_i\}_{i\in[\![1;2]\!]}$

$$\begin{aligned}
\mathcal{L}u_i^{(n+1)} &= f|_{\Omega_i} && \text{in } \Omega_i \\
u_i^{(n+1)} &= g && \text{on } \partial\Omega_i \setminus \Omega_j \\
\nabla u_i^{(n+1)} \cdot n_i + \mathcal{B}_i u_i^{(n+1)} &= -\nabla u_j^{(n)} \cdot n_j + \mathcal{B}_i u_j^{(n)} && \text{on } \partial\Omega_i \cap \Omega_j
\end{aligned}$$

Optimal boundary conditions

Given general BC operators $\{\mathcal{B}_i\}_{i\in[\![1;2]\!]}$

$$\begin{aligned} \mathcal{L}u_i^{(n+1)} &= f|_{\Omega_i} && \text{in } \Omega_i \\ u_i^{(n+1)} &= g && \text{on } \partial\Omega_i \setminus \Omega_j \\ \nabla u_i^{(n+1)} \cdot n_i + \mathcal{B}_i u_i^{(n+1)} &= -\nabla u_j^{(n)} \cdot n_j + \mathcal{B}_i u_j^{(n)} && \text{on } \partial\Omega_i \cap \Omega_j \end{aligned}$$

Use the Dirichlet-to-Neumann operators $\{\mathrm{DtN}_i\}_{i\in[\![1;2]\!]}$

$$\begin{aligned} \mathrm{DtN}_i \colon \partial\Omega_i \setminus \Omega &\to \mathbb{R} \\ g &\mapsto \nabla v \cdot n_i \end{aligned}$$

where v satisfies

$$\begin{aligned} \mathcal{L}v &= 0 \text{ in } \Omega_i \setminus \Omega_j \\ v &= 0 \text{ on } \partial\Omega_i \cap \partial\Omega \\ v &= g \text{ on } \partial\Omega_i \cap \partial\Omega_j \end{aligned}$$

Coarse space operators

Motivation

No exchange of global information

$$\kappa(M^{-1}A) \leqslant C\frac{1}{H^2}\left(1 + \frac{H}{\delta}\right)$$

- level of overlap δ
- characteristic size of a subdomain H

[Le Tallec 1994; Toselli and Widlund 2005]

Abstract setting

Let Z be a tall and skinny matrix

$$\begin{aligned} \text{search for } & \min_v ||A(y + Zv) - f||_{A^{-1}} \\ \iff & \min_v v^T Z^T A Z v - 2 v^T Z^T (f - Ay) + \lambda \\ \implies & v = \left(Z^T A Z\right)^{-1} Z^T (f - Ay) \end{aligned}$$

Abstract setting

Let Z be a tall and skinny matrix

$$\text{search for } \min_v ||A(y+Zv)-f||_{A^{-1}}$$

$$\iff \min_v v^T Z^T AZv - 2v^T Z^T(f-Ay) + \lambda$$

$$\implies v = \left(Z^T AZ\right)^{-1} Z^T(f-Ay)$$

Define a two-level additive Schwarz method as:

$$M^{-1} = Z\left(Z^T AZ\right)^{-1} Z^T + \sum_{i=1}^{N} R_i(R_i^T AR_i)^{-1} R_i^T$$

Nicolaides coarse space

Smallest coarse space possible [Nicolaides 1987]

$$Z = \begin{bmatrix} R_1^T D_1 R_1 & \cdots & R_N^T D_N R_N \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Nicolaides coarse space

Smallest coarse space possible [Nicolaides 1987]

$$Z = \begin{bmatrix} R_1^T D_1 R_1 & \cdots & R_N^T D_N R_N \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Galerkin operator $E = Z^T A Z$ of order $N \ll n$

Much better conditioning $\kappa(M^{-1}A) \leqslant C\left(1 + \frac{H}{\delta}\right)$

DtN coarse space

For this coarse space, $\mathcal{L} = \eta - \nabla \cdot (\kappa \nabla)$

- errors are harmonic inside subdomains
- fast decay $\implies$ large eigenvalue of the DtN operator

Solve local eigenvalue problems $\mathrm{DtN}_i(u) = \lambda \kappa u$

Substructuring methods

Algebraic decomposition

Given two subdomains, assume that:

$$A = \begin{bmatrix} A_{11} & 0 & A_{1\Gamma} \\ 0 & A_{22} & A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma} \end{bmatrix} \qquad f = \begin{bmatrix} f_1 \\ f_2 \\ f_\Gamma \end{bmatrix}$$

Now, eliminate x_1 and x_2 from $Ax = f$:

$$\begin{aligned} \left(A_{\Gamma\Gamma} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma}\right) x_\Gamma &= f_\Gamma - A_{\Gamma 1}A_{11}^{-1}f_1 - A_{\Gamma 2}A_{22}^{-1}f_2 \\ &= g_b = g_b^{(1)} + g_b^{(2)} \end{aligned}$$

The Schur complement S_p is defined as:

$$\begin{aligned} S_p &= A_{\Gamma\Gamma}^{(1)} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma} + A_{\Gamma\Gamma}^{(2)} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma} \\ &= S_p^{(1)} + S_p^{(2)} \end{aligned}$$

Solving the condensed system

Appropriate preconditioner for $S_p x_b = g_b$?

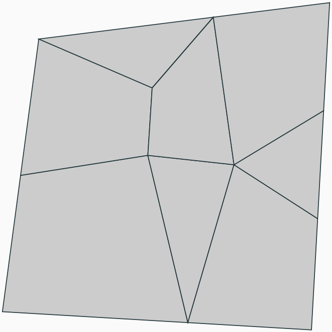
If $S_p^{(1)} = S_p^{(2)}$, choose $M^{-1} = \frac{1}{4}\left({S_p^{(1)}}^{-1} + {S_p^{(2)}}^{-1}\right)$

$$
\begin{aligned}
A^{(1)} &= \begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma}^{(1)} \end{bmatrix} \\
&= \begin{bmatrix} I & 0 \\ A_{\Gamma 1}{A_{11}}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & S_p^{(1)} \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{1\Gamma} \\ 0 & I \end{bmatrix}
\end{aligned}
$$

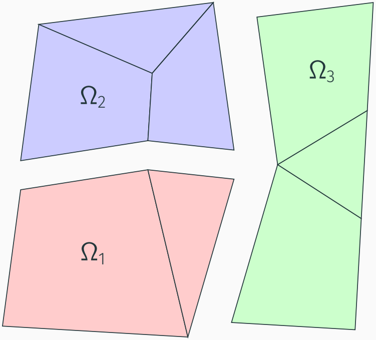
Thus,

$$
{S_p^{(1)}}^{-1}\lambda = \begin{bmatrix} 0 & I \end{bmatrix} {A^{(1)}}^{-1} \begin{bmatrix} 0 & 1 \end{bmatrix}^T \lambda
$$

Back to domain decomposition

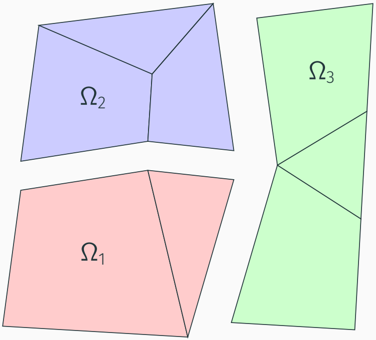


[Gosselet and Rey 2006]

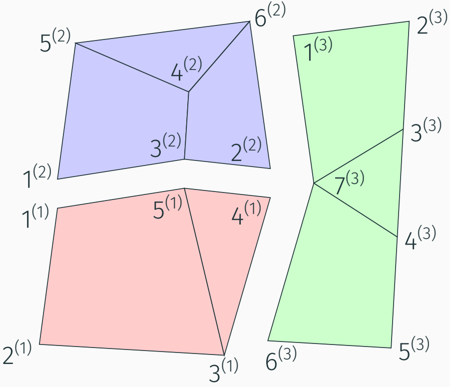


Subdomain tearing

Back to domain decomposition



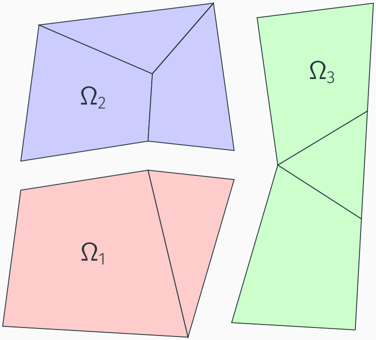
[Gosselet and Rey 2006]

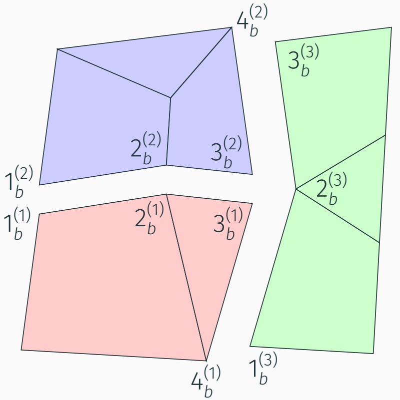


Local numbering

$$A^{(k)} = \begin{bmatrix} A_{ii} & A_{ib} \\ A_{bi} & A_{bb} \end{bmatrix}$$

Back to domain decomposition



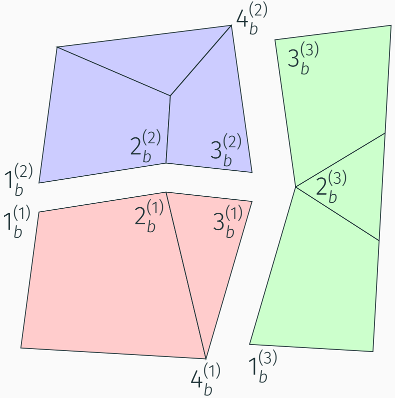


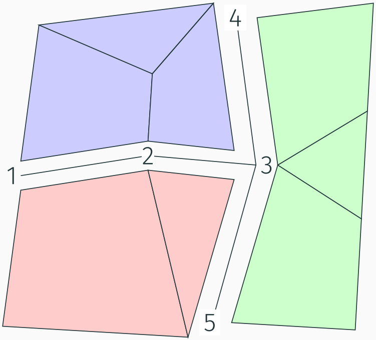
[Gosselet and Rey 2006]

Elimination of interior d.o.f.

$$S_p^{(k)} = A_{bb} - A_{bi}A_{ii}^{-1}A_{ib}$$

Back to domain decomposition

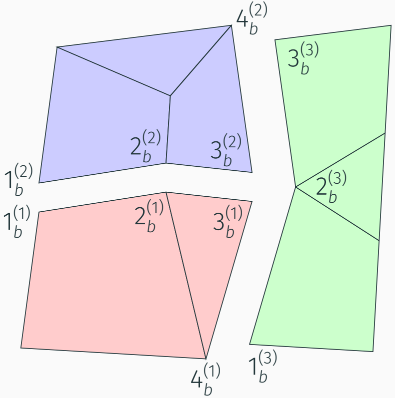


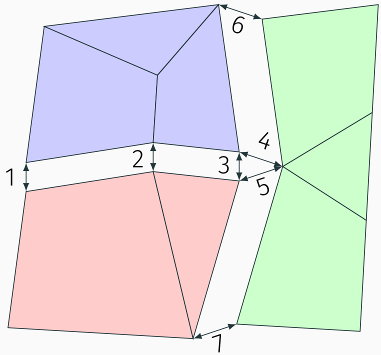


Jump operators: $\{B^{(i)}\}_{i=1}^{3}$

Primal constraints
[Mandel 1993]

Back to domain decomposition





Jump operators: $\{\underline{B}^{(i)}\}_{i=1}^{3}$

Dual constraints
[Farhat and Roux 1991]

Condensed system

The new system reads:

$$\forall k \in [\![1; N]\!], S_p^{(k)} x_b^{(k)} = g_b^{(k)} + \lambda_b^{(k)}$$

Condensed system

The new system reads:

$$\forall k \in [\![1; N]\!], S_p^{(k)} x_b^{(k)} = g_b^{(k)} + \lambda_b^{(k)}$$

$$\sum_{k=1}^{N} \underline{B}^{(k)} x_b^{(k)} = 0$$

Condensed system

The new system reads:

$$\forall k \in [\![1; N]\!], S_p^{(k)} x_b^{(k)} = g_b^{(k)} + \lambda_b^{(k)}$$

$$\sum_{k=1}^{N} \underline{B}^{(k)} x_b^{(k)} = 0$$

$$\sum_{k=1}^{N} B^{(k)} \lambda_b^{(k)} = 0$$

Condensed system

The new system reads:

$$
\begin{aligned}
\forall k \in [\![1; N]\!], S_p^{(k)} x_b^{(k)} &= g_b^{(k)} + \lambda_b^{(k)} \\
{R_b^{(k)}}^T \lambda_b^{(k)} &= 0 \\
\sum_{k=1}^{N} \underline{B}^{(k)} x_b^{(k)} &= 0 \\
\sum_{k=1}^{N} B^{(k)} \lambda_b^{(k)} &= 0
\end{aligned}
$$

Primal methods

Define a unique displacement $x_b \implies x_b^{(k)} = {B^{(k)}}^T x_b$

Eliminate the reactions:

$$\sum_{k=1}^{N} B^{(k)} S_p^{(k)} {B^{(k)}}^T x_b = \sum_{k=1}^{N} B^{(k)} g^{(k)}$$

Weighted sum of pseudo-inverses as a preconditioner:

$$M^{-1} = \sum_{k=1}^{N} B^{(k)} D_p^{(k)} {S_p^{(k)}}^{\dagger} D_p^{(k)} {B^{(k)}}^T$$

The preconditioner should be applied to vectors in $\mathrm{Im}(S_p)$

Balancing Domain Decomposition

A vector r_b is balanced if:

$$\sum_{k=1}^{N} R_b^{(k)^T} D_p^{(k)} B^{(k)^T} r_b = 0$$

Let

$$\mathsf{R}_b = \left[B^{(1)} D_p^{(1)} R_b^{(1)} \quad \cdots \quad B^{(N)} D_p^{(N)} R_b^{(N)} \right]$$

$$\mathsf{S}_p = \sum_{k=1}^{N} B^{(k)} S_p^{(k)} B^{(k)^T}$$

Then, $P = I - \mathsf{R}_b \left(\mathsf{R}_b^T \mathsf{S}_p \mathsf{R}_b \right)^{-1} \mathsf{R}_b^T \mathsf{S}_p \implies \mathsf{R}_b^T \mathsf{S}_p P = 0$

Define a unique reaction $\lambda_b \implies \lambda_b^{(k)} = \underline{B}^{(k)^T} \lambda_b$

Eliminate the displacements:

$$\forall k \in [\![1; N]\!], x_b^{(k)} = S_d^{(k)}(g_b^{(k)} + \underline{B}^{(k)^T}\lambda_b) + R_b^{(k)}\alpha^{(k)}$$

$$0 = R_b^{(k)^T}(g_b^{(k)} + \underline{B}^{(k)^T}\lambda_b)$$

With similar notations:

$$\begin{bmatrix} \mathsf{S}_d & \underline{\mathsf{R}}_b \\ \underline{\mathsf{R}}_b^T & 0 \end{bmatrix} \begin{bmatrix} \lambda_b \\ \alpha \end{bmatrix} = \begin{bmatrix} -\mathbf{b}_d \\ -\mathbf{g}_b \end{bmatrix}$$

Implementation aspects

Distributed parallelism

Domain decomposition widely use with FE

Easy way to speedup the assembly phase

Message Passing Interface

- distributed numbering of unknowns
- ghost elements for assembled operators
- connectivity with neighboring subdomains

Matrix–vector products

- for overlapping Schwarz methods:

$$\forall k \in [\![1; N]\!], R_k A x = R_k \sum_{i=1}^{N} A R_i^T D_i R_i x$$
$$= \sum_{i=1}^{N} R_k A R_i^T D_i x_i$$

- for substructuring methods:

$$\forall k \in [\![1; N]\!], {B^{(k)}}^T \mathsf{S}_p x_b = {B^{(k)}}^T \sum_{j=1}^{N} B^{(j)} S_p^{(j)} {B^{(j)}}^T x_b$$

Dot products

Same strategy using the appropriate scaling:

$$\begin{aligned}(u, v) &= u^T v \\ &= u^T \sum_{k=1}^{N} R_k^T D_k R_k v \\ &= \sum_{k=1}^{N} u_k^T D_k v_k\end{aligned}$$

Coarse space corrections

Implies global information exchanges:

- local computations
- all-to-few gathering
- either compute or idle
- few-to-all scattering
- local computations

Links with multigrid methods

Convergence results

The damped Jacobi preconditioner reads $M^{-1} = \omega D^{-1}$

Error propagation:

$$\begin{aligned} e_{k+1} &= (I - M^{-1}A)e_k \\ &= (I - M^{-1}A)^{k+1}e_0 \end{aligned}$$

On a uniform 1D grid:

$$(I - M^{-1}A) = I - \frac{\omega}{2}\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \\ & & -1 & 2 \end{bmatrix},$$

whose eigenvalues are $\left\{\mu_k = 1 - 2\omega\sin^2\left(\frac{k\pi}{2n}\right)\right\}_{k\in[\![1;n]\!]}$

Why use multiple grids?

Low frequencies of the error on a coarser grid:

- more oscillatory
- cheaper to relax
- better condition number of the operator

+ aliasing for the high frequencies

A two-grid method

Given x_0,

- smooth ν times to relax $Ax_k = f$ with the initial guess x_k
- compute the residual $r_k = f - Ax_k$
- restrict the residual to the coarser grid R_k
- solve $A_C E_{k+1} = R_k$
- interpolate the error back on the fine grid e_{k+1}
- smooth μ times with the initial guess $x_k + e_{k+1}$

Restriction and prolongation operators

- restriction R^h_{2h}: canonical injection, full-weighting $R^h_{2h,\text{fw}}$
- prolongation P^{2h}_h: interpolation
- coarse operator $A_C = A_{2h}$

On a regular grid,

$$
\begin{aligned}
2R^h_{2h,\text{fw}} &= {P^{2h}_h}^T \\
A_{2h} &= R^h_{2h,\text{fw}} A_h P^{2h}_h
\end{aligned}
$$

Iteration matrix of the two-grid method

- coarse grid correction $e_{k+1} = P_h^{2h}\left(R_{2h}^h A_h P_h^{2h}\right)^{-1} R_{2h}^h A_h e_k$
- coarse grid operator

$$
\begin{aligned}
\tilde{x}_{k+1} &= \tilde{x}_k + e_{k+1} \\
&= \left(I - P_h^{2h}\left(R_{2h}^h A_h P_h^{2h}\right)^{-1} R_{2h}^h A_h\right)\tilde{x}_k \\
&\quad + P_h^{2h}\left(R_{2h}^h A_h P_h^{2h}\right)^{-1} R_{2h}^h f
\end{aligned}
$$

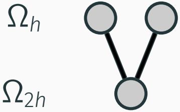
- complete iteration

$$
x_{k+1} = S^\mu\left(I - P_h^{2h}\left(R_{2h}^h A_h P_h^{2h}\right)^{-1} R_{2h}^h A_h\right) S^\nu x_k + g_k
$$

Recursion to create different cycles

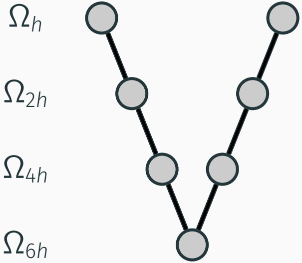
Solve the coarse problem using a two-grid method

Two-grid cycle



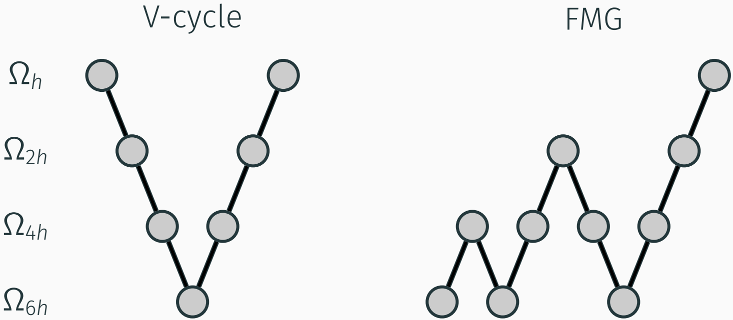
Solve the coarse problem using a two-grid method

V-cycle



Recursion to create different cycles

Solve the coarse problem using a two-grid method



Remarks

Multigrid methods may be used as preconditioners
Use the cycles to solve $M^{-1}z = r$, with 0 as an initial guess

Algebraic multigrid (AMG) is a black-box solver
Build the restriction + prolongation operators on the fly