An introduction to domain decomposition methods

Victorita Dolean

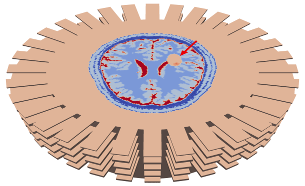
with: F. Nataf, P. Jolivet, P.-H. Tournier

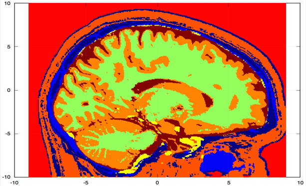
LJLL French-Spanish summer school,
Ciudad Real, 7-11 July 2025

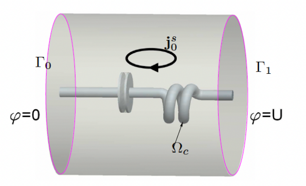
Outline

Introduction

EM wave propagation in heterogeneous media







Maxwell's equations

Reconstruct the permittivity ε

$$\nabla \times (\mu^{-1}\nabla \times \mathbf{E}) - \omega^2 \varepsilon \mathbf{E} = \mathbf{J}$$

- $\mathbf{E}$ is the electric field
- $\mu > 0$ is the magnetic permeability
- $\varepsilon > 0$ is the electric permittivity.
- ω is the frequency

Challenges

- **High frequency**: solution highly oscillatory $\rightsquigarrow$ pollution effect, large linear systems.
- **Low frequency**: near singular operator with a huge near kernel.

AIM

The linear system inherits the properties of the PDE $\rightsquigarrow$ design a **robust** solver.

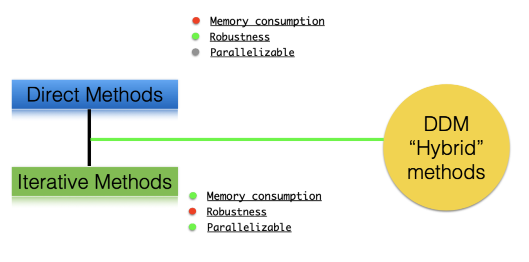
$Au = b$? Landscape of linear solvers

Iterative Methods

- Fixed-point: Jacobi, Gauss-Seidel, SSOR
- Krylov methods:
  - Conjugate Gradient (Stiefel–Hestenes)
  - GMRES (Y. Saad), QMR (R. Freund)
  - MinRes, BiCGSTAB (van der Vorst)

Direct Solvers

- **MUMPS** (J.Y. L'Excellent)
- **SuperLU** (Demmel et al), **PastiX**
- **UMFPACK**, **PARDISO** (O. Schenk)



"Hybrid" Methods (DDM)

- Multigrid: Brandt, Ruge-Stüben, Falgout, McCormick, Ruhe, Notay . . .
- Domain decomposition (DDM): Widlund, Farhat, Mandel, Lions . . .

Natural parallel compromise between robustness and scalability

Sparse Gaussian Elimination: Complexity and Practical Limits

Asymptotic Complexity for Structured PDE Matrices

Method	**1D** ($d = 1$)	**2D** ($d = 2$)	**3D** ($d = 3$)
Dense matrix	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
Band structure exploited	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^{7/3})$
Sparse (e.g. nested dissection)	$\mathcal{O}(n)$	$\mathcal{O}(n^{3/2})$	$\mathcal{O}(n^2)$

Practical Limits (2025)

- Sparse direct solvers handle up to $n \sim 10^7$ in 2D and $n \sim 10^5$ in 3D on modern hardware.
- Fill-in in 3D limits scalability: memory and factorization time dominate.
- Hybrid methods (e.g. domain decomposition or multilevel solvers) are preferred for large-scale problems.

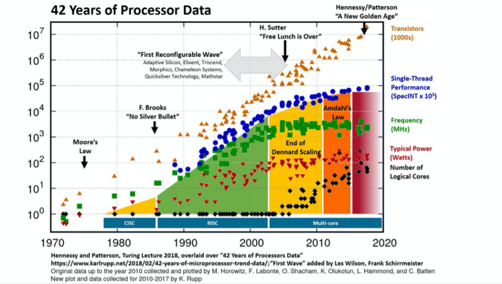
Modern Sparse Direct Solvers

- **PARDISO**: High-performance sparse solver - `https://www.pardiso-project.org`
- **SuperLU (Dist)**: Parallel solver for general matrices- `https://github.com/xiaoyeli/superlu_dist`
- **MUMPS**: MPI-parallel multifrontal solver - `http://mumps-solver.org`
- **UMFPACK** (SuiteSparse): Serial sparse LU - `https://suitesparse.com`

Need & Opportunities for massively parallel computing

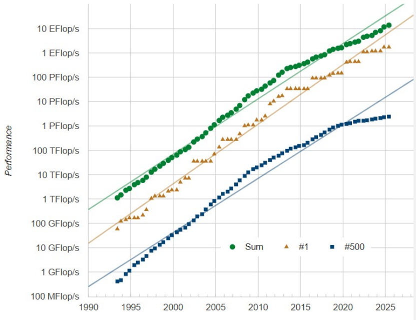
Processor Evolution (1970–2020)

- Moore's Law: steady increase in transistor count until today.
- Dennard Scaling breakdown (2005): power and frequency hit physical limits.
- Led to multi-core architectures, performance stagnation in single-thread execution.
- **New focus:** parallelism, heterogeneity, and architectural innovation.



Top500 Supercomputers (1993–2025)

- Exponential performance growth of top systems (1), 500 and total peak.
- Entry point into the **exascale era** reached around 2023–2024.
- Reflects hardware scaling, algorithmic advances, and parallelism.



Need & Opportunities for massively parallel computing

The Rise of Parallel Machines

Parallel computing is now accessible to everyone:

- Laptops (Apple Mx, Linux, Windows): 4–12 cores
- Desktops/Workstations: 16–128 cores
- Lab Clusters: ∼300 cores
- University HPC Clusters: ∼10,000 cores
- Cloud Infrastructures (AWS, Azure): elastic, on-demand compute
- National Supercomputers: $> 100,000$ cores, e.g., Fugaku, LUMI, Frontier

All fields of science and engineering are affected.

Hardware Trends (2025)

- ARM-based SoCs: **A64FX** (Fugaku), **Apple M1/M2**, **Graviton3**
- Scalable Vector Extensions (SVE) enable efficient HPC+AI workloads
- High Bandwidth Memory (HBM) improves memory-bound performance
- Heterogeneous systems: CPUs + GPUs + AI accelerators

Software Evolution

- **Languages**: Julia, Rust, Python (MPI4Py), offer easy parallelism
- **Libraries**: PETSc, HPDDM, ScaLAPACK (linear algebra); DUNE, OpenFOAM (PDEs)
- **Standards**: MPI + OpenMP + OpenACC remain critical

The First Domain Decomposition Method

The Original Schwarz Method (H.A. Schwarz, 1870)

Solves $-\Delta u = f$ in Ω with $u = 0$ on $\partial\Omega$ using overlapping subdomains:

Iteration Scheme: Schwarz Alternating Method

Given u_1^n, u_2^n, compute:

$$\begin{aligned} -\Delta u_1^{n+1} &= f \quad \text{in } \Omega_1 \\ u_1^{n+1} &= 0 \quad \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} &= u_2^n \quad \text{on } \partial\Omega_1 \cap \overline{\Omega}_2 \end{aligned}$$

$$\begin{aligned} -\Delta u_2^{n+1} &= f \quad \text{in } \Omega_2 \\ u_2^{n+1} &= 0 \quad \text{on } \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} &= u_1^{n+1} \quad \text{on } \partial\Omega_2 \cap \overline{\Omega}_1 \end{aligned}$$

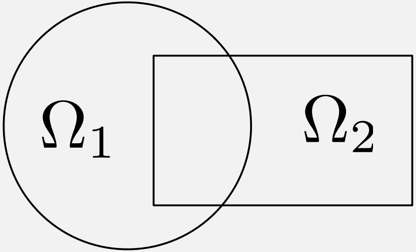


Illustration: two overlapping subdomains Ω_1, Ω_2

Key Characteristics

- Naturally parallel, but convergence is very slow.
- Requires **overlap** between subdomains.
- The parallel version is known as the **Jacobi–Schwarz Method (JSM)**.

Continuous ASM and RAS – Key Ingredients

Local-to-Global Strategy

We solve on local functions u_i supported on Ω_i and reconstruct the global function u.

Extension Operators

Each E_i extends a local function $w_i : \Omega_i \to \mathbb{R}$ to $E_i(w_i) : \Omega \to \mathbb{R}$ by zero outside Ω_i.

Partition of Unity

Let $\chi_i : \Omega_i \to \mathbb{R}$ satisfy:

$$\chi_i \geq 0, \quad \chi_i(x) = 0 \text{ on } \partial\Omega_i, \quad \text{and} \quad w(x) = \sum_{i=1}^{2} E_i(\chi_i \, w|_{\Omega_i}).$$

Iteration

Given $u^n \approx u$, compute u^{n+1} by solving local problems and combining them using E_i and χ_i.

Continuous ASM and RAS – Local Solves and Gluing

Local Subproblems (for $i = 1, 2$)

At each iteration n, solve:

$$\begin{aligned} -\Delta u_i^{n+1} &= f \quad \text{in } \Omega_i \\ u_i^{n+1} &= 0 \quad \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i^{n+1} &= u^n \quad \text{on } \partial\Omega_i \cap \overline{\Omega}_{3-i} \end{aligned}$$

Pros & Cons

- RAS: faster convergence, non-symmetric
- ASM: easier analysis, symmetric
- Both methods are **parallel**, overlap-dependent

Gluing the Solutions: Two Strategies

Restricted Additive Schwarz (RAS)

$$u^{n+1} = \sum_{i=1}^{2} E_i(\chi_i\, u_i^{n+1})$$

Uses partition of unity for overlap control.

Additive Schwarz (ASM)

$$u^{n+1} = \sum_{i=1}^{2} E_i(u_i^{n+1})$$

Pure sum of extensions (no weighting).

Classical Jacobi Method

Linear System

$$\mathbf{A}\mathbf{U} = \mathbf{F}, \quad A \in \mathbb{R}^{m\times m}, \quad \mathbf{U}, \mathbf{F} \in \mathbb{R}^{m}$$

Jacobi Iteration

Let $D = \mathrm{diag}(A)$ (diagonal of A). Then:

$$\mathbf{U}^{n+1} = \mathbf{U}^{n} + D^{-1}\left(\mathbf{F} - A\mathbf{U}^{n}\right) = \mathbf{U}^{n} + D^{-1}\mathbf{r}^{n}$$

where $\mathbf{r}^{n} = \mathbf{F} - A\mathbf{U}^{n}$ is the residual.

Key Features

- Simple, parallelizable update (diagonal inverse).
- Convergence only under specific conditions (e.g., diagonal dominance).

Block Jacobi: Domain Partitioning

Index Partition

Split degrees of freedom into two index sets:

$$\mathcal{N}_1 := \{1, \ldots, m_s\}, \quad \mathcal{N}_2 := \{m_s + 1, \ldots, m\}$$

Define:

$$\mathbf{U}_1 := \mathbf{U}_{|\mathcal{N}_1}, \quad \mathbf{U}_2 := \mathbf{U}_{|\mathcal{N}_2}, \quad \mathbf{F}_1, \mathbf{F}_2 \text{ likewise.}$$

Block Form of A

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{pmatrix}$$

Block Jacobi Iteration — Local Systems

Block Jacobi Update

Solve the block systems:

$$A_{11}\mathbf{U}_1^{n+1} = \mathbf{F}_1 - A_{12}\mathbf{U}_2^n$$
$$A_{22}\mathbf{U}_2^{n+1} = \mathbf{F}_2 - A_{21}\mathbf{U}_1^n$$

Matrix View

$$\begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^{n+1} \\ \mathbf{U}_2^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1 - A_{12}\mathbf{U}_2^n \\ \mathbf{F}_2 - A_{21}\mathbf{U}_1^n \end{pmatrix}$$

Block Jacobi — Residual-Based Formulation

Compact Notation

Let $\mathbf{U}^n = (\mathbf{U}_1^n, \mathbf{U}_2^n)^T$. Then:

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}^{-1} \left(\mathbf{F} - A\mathbf{U}^n\right)$$

Interpretation

- Perform local solves on each block (A_{11}, A_{22}).
- Apply local inverses to the residual.
- **Parallel**: no coupling in the matrix used for the update.

Block-Jacobi Method — Compact Matrix Form

Operators

- R_1, R_2: Restriction operators from $\mathcal{N}$ to $\mathcal{N}_1$, $\mathcal{N}_2$ respectively
- R_1^T, R_2^T: Extension operators
- $A_i = R_i A R_i^T$: local block matrix

Residual

$$\mathbf{r}^n = \mathbf{F} - A\mathbf{U}^n, \quad \mathbf{r}_i^n = \mathbf{r}^n_{|\mathcal{N}_i}$$

Compact Block-Jacobi Update

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \left(R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2 \right) \mathbf{r}^n$$

- Parallel update using local block solves
- A_1^{-1} and A_2^{-1} are local inverses (independent)
- Structure generalizes to N blocks trivially

Schwarz Methods as Block-Jacobi Algorithms (1D Case)

Problem Setup

Let $\Omega = (0,1)$ with Dirichlet BCs:

$$-\Delta u = f \text{ in } \Omega, \quad u(0) = u(1) = 0$$

Discretize with m internal nodes using 3-point finite differences:

$$A\mathbf{U} = \mathbf{F}, \quad A \in \mathbb{R}^{m \times m}$$

where:

$$A_{jj} = \frac{2}{h^2}, \quad A_{j,j\pm1} = -\frac{1}{h^2}, \quad h = \frac{1}{m+1}$$

Subdomain Decomposition

Overlap of width h:

$$\Omega_1 = (0, (m_s + 1)h), \quad \Omega_2 = (m_s h, 1)$$

Jacobi–Schwarz Update on Ω_1

$$\begin{cases} -\dfrac{u_{1,j-1}^{n+1} - 2u_{1,j}^{n+1} + u_{1,j+1}^{n+1}}{h^2} = f_j, \quad 1 \le j \le m_s \\ u_{1,0}^{n+1} = 0 \\ u_{1,m_s+1}^{n+1} = u_{2,m_s+1}^{n} \end{cases}$$

Matrix Formulation

$$A_{11}\mathbf{U}_1^{n+1} + A_{12}\mathbf{U}_2^{n} = \mathbf{F}_1$$
$$A_{22}\mathbf{U}_2^{n+1} + A_{21}\mathbf{U}_1^{n} = \mathbf{F}_2$$

Schwarz = Block-Jacobi (Under Minimal Overlap)

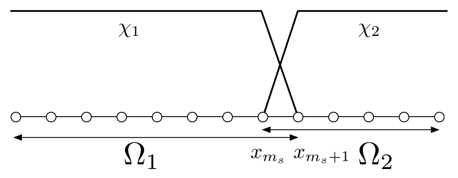
Extension Operators

$$E_1(U_1) = \begin{pmatrix} U_1 \\ 0 \end{pmatrix}, \quad E_2(U_2) = \begin{pmatrix} 0 \\ U_2 \end{pmatrix}$$

Partition of Unity

With overlap:

$$E_1(U_1)+E_2(U_2) = E_1(\chi_1 U_1)+E_2(\chi_2 U_2) = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix}$$



Key Insight

When the overlap is minimal, the discrete forms of:

- Additive Schwarz (AS)
- Restricted Additive Schwarz (RAS)
- Jacobi-Schwarz (JS)

all reduce to the same block-Jacobi method.

From Continuous to Discrete: Domain Decomposition View

Continuous Level

- Domain: $\Omega = \bigcup_{i=1}^{N} \Omega_i$ (overlapping decomposition)
- Global function: $u : \Omega \to \mathbb{R}$
- Restriction: $u_i = u|_{\Omega_i}$
- Extension: $E_i(u_i) : \Omega \to \mathbb{R}$ (zero outside Ω_i)
- Partition of unity: functions $\chi_i : \Omega_i \to \mathbb{R}$ with

$$u = \sum_{i=1}^{N} E_i(\chi_i \, u|_{\Omega_i})$$

Discrete Level

- Degrees of freedom: $\mathcal{N} = \bigcup_{i=1}^{N} \mathcal{N}_i$
- Global vector: $U \in \mathbb{R}^{\#\mathcal{N}}$
- Restriction: $R_i \in \{0, 1\}^{\#\mathcal{N}_i \times \#\mathcal{N}}$
- Extension: R_i^T (transpose of R_i)
- Partition of unity: diagonal matrices D_i with positive entries s.t.:

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

Restriction Operators in Finite Element Decomposition

Mesh-Based Domain Decomposition

- $\mathcal{T}_h$: global mesh of domain Ω
- $\mathcal{T}_{h,i}$: mesh of subdomain Ω_i
- V_h: global finite element space
- $V_{h,i}$: local FE space on $\mathcal{T}_{h,i}$
- u_h: global FE solution

Restriction Operator

Restriction of u_h to Ω_i:

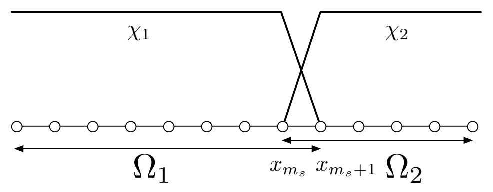
$$r_i(u_h) = u_h|_{\Omega_i}, \quad r_i : V_h \to V_{h,i}$$

Matrix form:

$$R_i : \mathbb{R}^{\#\mathcal{N}} \to \mathbb{R}^{\#\mathcal{N}_i}$$

(R_i is a Boolean selector)

Geometric Interpretation



Algebraic Form: Boolean Matrix R_i

$$R_i = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \end{bmatrix}$$

Each row selects one DOF from $\mathcal{N}$.

Operators

- Restriction: $R_i : \mathbb{R}^{\#\mathcal{N}} \to \mathbb{R}^{\#\mathcal{N}_i}$
- Prolongation (extension): $R_i^T : \mathbb{R}^{\#\mathcal{N}_i} \to \mathbb{R}^{\#\mathcal{N}}$
- Local matrices: $A_i = R_i A R_i^T$
- Partition of unity:

$$D_i \in \mathbb{R}^{\#\mathcal{N}_i \times \#\mathcal{N}_i} \text{ diagonal, s.t.} \quad \sum_{i=1}^{N} R_i^T D_i R_i = I$$

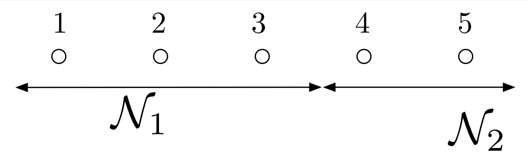
Purpose

These ingredients allow expressing local computations and gluing into a global update.

Two-Subdomain Example: Finite Differences (No Overlap)

Domain Decomposition

$$\mathcal{N} = \{1, 2, 3, 4, 5\}, \quad \mathcal{N}_1 = \{1, 2, 3\}, \quad \mathcal{N}_2 = \{4, 5\}$$



Restriction Matrices

$$R_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

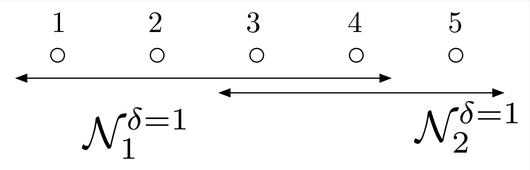
Partition of Unity

$$D_1 = I_{3\times 3}, \quad D_2 = I_{2\times 2}$$

Two-Subdomain Example: Finite Differences (With Overlap)

Overlapping Decomposition

$$\mathcal{N}_1^{\delta=1} = \{1, 2, 3, 4\}, \quad \mathcal{N}_2^{\delta=1} = \{3, 4, 5\}$$



Restriction Matrices

$$R_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

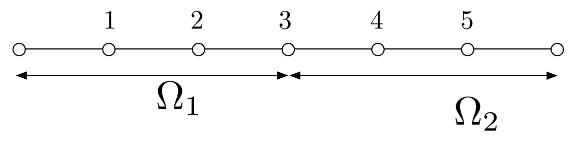
Partition Matrices

$$D_1 = \mathrm{diag}(1, 1, \tfrac{1}{2}, \tfrac{1}{2}), \quad D_2 = \mathrm{diag}(\tfrac{1}{2}, \tfrac{1}{2}, 1)$$

Two-Subdomain Example: Finite Elements (Overlap)

FE Node Sets

$$\mathcal{N}_1 = \{1, 2, 3\}, \quad \mathcal{N}_2 = \{3, 4, 5\}$$



Restriction Matrices

$$R_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

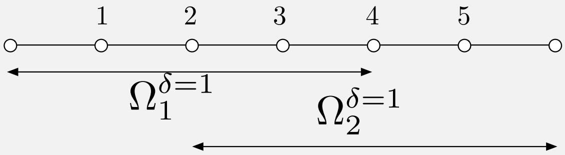
Partition Matrices

$$D_1 = \mathrm{diag}(1, 1, \tfrac{1}{2}), \quad D_2 = \mathrm{diag}(\tfrac{1}{2}, 1, 1)$$

Two-Subdomain Example: FE with Overlapping Partition (Extended)

FE Node Sets

$$\mathcal{N}_1^{\delta=1} = \{1,2,3,4\}, \quad \mathcal{N}_2^{\delta=1} = \{2,3,4,5\}$$



Restriction Matrices

$$R_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Partition Matrices

$$D_1 = \mathrm{diag}(1, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}), \; D_2 = \mathrm{diag}(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, 1)$$

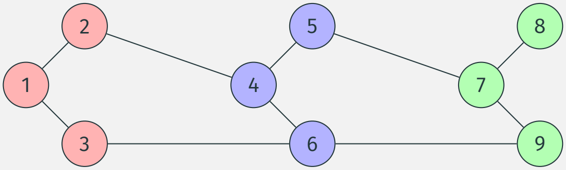
Multi-Domain Partitioning: General Procedure

Graph-Based Partitioning (METIS, SCOTCH)

- From matrix A, construct a graph G:
  - Nodes $\leftrightarrow$ degrees of freedom
  - Edges: $A_{ij} \neq 0$
- Symmetrize G if needed
- Apply partitioner to divide $\mathcal{N}$ into N subdomains

Partitioning Goals

- Load balancing: equal work per subdomain
- Minimal communication: few cross edges



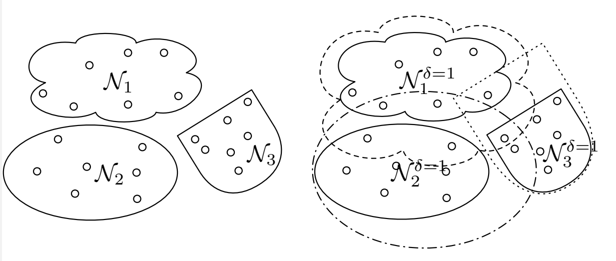
Graph G partitioned into 3 subdomains

Multi-D Algebraic Setting with Overlap

Overlap Construction

- From disjoint sets $\mathcal{N}_i$, define overlapping sets:

$$\mathcal{N}_i^{\delta=1} = \mathcal{N}_i \cup \text{neighbors of } \mathcal{N}_i$$



Algebraic Partition of Unity

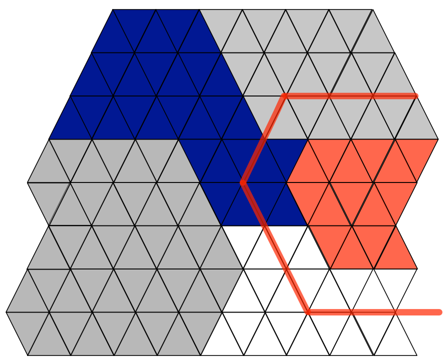
Let R_i be the restriction from $\mathcal{N}$ to $\mathcal{N}_i^{\delta=1}$.

$$(D_i)_{jj} = \frac{1}{\#\mathcal{M}_j}, \quad \mathcal{M}_j := \{i : j \in \mathcal{N}_i^{\delta=1}\}$$

Multi-D Algebraic Finite Element Decomposition

Mesh and Overlap

- Let $\mathcal{T}_h$: mesh of Ω
- Each $\mathcal{T}_{h,i}$ gives overlapping Ω_i



FE Basis and Index Partition

Let $\{\phi_k\}_{k\in\mathcal{N}}$ be basis functions:

$$\mathcal{N}_i := \{k : \operatorname{supp}(\phi_k) \cap \Omega_i \neq \emptyset\}$$

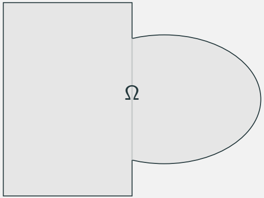
Multiplicity per node:

$$\mu_k := \#\{j : \operatorname{supp}(\phi_k) \cap \Omega_j \neq \emptyset\}$$

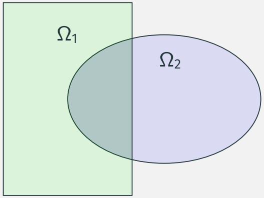
Algebraic Partition of Unity

$$(D_i)_{kk} = \frac{1}{\mu_k}, \quad k \in \mathcal{N}_i$$

Let the discretised Poisson problem: $A\mathbf{U} = \mathbf{F} \in \mathbb{R}^n$.



Let the discretised Poisson problem: $A\mathbf{U} = \mathbf{F} \in \mathbb{R}^n$. Given a decomposition of $[\![1; n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define: the restriction operator R_i from $\mathbb{R}^{[\![1;n]\!]}$ into $\mathbb{R}^{\mathcal{N}_i}$, R_i^T as the extension by 0 from $\mathbb{R}^{\mathcal{N}_i}$ into $\mathbb{R}^{[\![1;n]\!]}$.

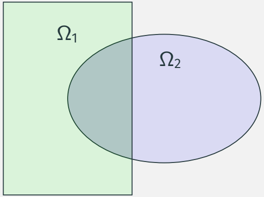


Summary: Additive Schwarz

Let the discretised Poisson problem: $A\mathbf{U} = \mathbf{F} \in \mathbb{R}^n$. Given a decomposition of $[\![1;n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define: the restriction operator R_i from $\mathbb{R}^{[\![1;n]\!]}$ into $\mathbb{R}^{\mathcal{N}_i}$, R_i^T as the extension by 0 from $\mathbb{R}^{\mathcal{N}_i}$ into $\mathbb{R}^{[\![1;n]\!]}$.
Find $\mathbf{U}^m \longrightarrow \mathbf{U}^{m+1}$ by solving concurrently:

$$\mathbf{U}_j^{m+1} = \mathbf{U}_j^m + A_j^{-1} R_j (\mathbf{F} - A\mathbf{U}^m),\ j = 1, 2$$

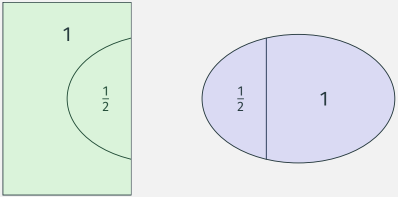
where $\mathbf{U}_i^m = R_i \mathbf{U}^m$ and $A_i := R_i A R_i^T$.



We have effectively divided, but we have yet to conquer.

Duplicated unknowns coupled via a partition of unity:

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

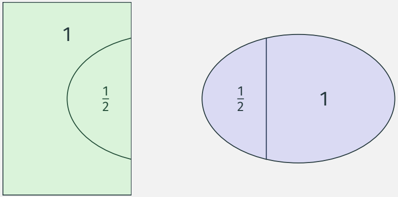


An introduction to Additive Schwarz II

We have effectively divided, but we have yet to conquer.

Duplicated unknowns coupled via a partition of unity:

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

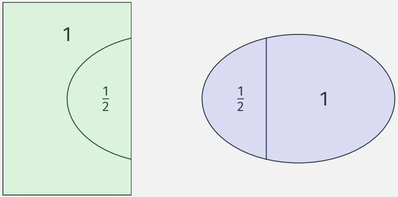


Then, $\mathbf{U}^{m+1} = \sum_{i=1}^{N} R_i^T D_i \mathbf{U}_i^{m+1}$.

We have effectively divided, but we have yet to conquer.

Duplicated unknowns coupled via a partition of unity:

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



Then, $\mathbf{U}^{m+1} = \sum_{i=1}^{N} R_i^T D_i \mathbf{U}_i^{m+1}.$ $\qquad M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i.$

RAS algorithm (Cai & Sarkis, 1999)

Algebraic RAS Formulation: Equivalence and Use

RAS as Global Iteration

The RAS method updates the global iterate using local solves and partition weights:

$$\boxed{\mathbf{U}^{n+1} = \mathbf{U}^{n} + M_{RAS}^{-1}\mathbf{r}^{n}, \quad \mathbf{r}^{n} := \mathbf{F} - A\mathbf{U}^{n}}$$

Local to Global Representation

The current iterate $\mathbf{U}^n$ is reconstructed from local components:

$$\boxed{\mathbf{U}^{n} = R_1^T D_1 \mathbf{U}_1^n + R_2^T D_2 \mathbf{U}_2^n}$$

Each $\mathbf{U}_i^n$ solves a local problem over an overlapping subdomain.

Krylov Usage

The operator M_{RAS}^{-1} serves as a preconditioner in Krylov subspace methods such as GMRES, BiCGStab, and others. This improves convergence for non-symmetric and ill-conditioned problems.

Preconditioners Summary

- **RAS (Restricted Additive Schwarz):**

$$M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i$$

- **ASM (Additive Schwarz Method):**

$$M_{ASM}^{-1} = \sum_{i=1}^{N} R_i^T A_i^{-1} R_i$$

- **SORAS (Symmetrized Overlapping RAS):**

$$M_{SORAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i B_i^{-1} D_i R_i$$

1D Domain Setup and Subdomain Error Equations

Domain Setup

Let $L > 0$, $\Omega = (0, L)$ split into:

- $\Omega_1 := (0, L_1)$
- $\Omega_2 := (l_2, L)$ with $l_2 \leq L_1$

Error: $e_i^n := u_i^n - u_{|\Omega_i}$

Affine solutions

$$e_1^{n+1}(x) = e_2^n(L_1) \cdot \frac{x}{L_1}$$

$$e_2^{n+1}(x) = e_1^{n+1}(l_2) \cdot \frac{L - x}{L - l_2}$$

Subdomain Error PDEs

e_1^{n+1} in Ω_1:

$$-\frac{d^2 e_1^{n+1}}{dx^2} = 0$$
$$e_1^{n+1}(0) = 0$$
$$e_1^{n+1}(L_1) = e_2^n(L_1)$$

e_2^{n+1} in Ω_2:

$$-\frac{d^2 e_2^{n+1}}{dx^2} = 0$$
$$e_2^{n+1}(l_2) = e_1^{n+1}(l_2)$$
$$e_2^{n+1}(L) = 0$$

Convergence Condition and Visualization

Interface Coupling

$$e_2^{n+1}(L_1) = e_2^n(L_1) \cdot \frac{l_2}{L_1} \cdot \frac{L - L_1}{L - l_2}$$

Observation

The current interface value depends linearly on the previous iterate.

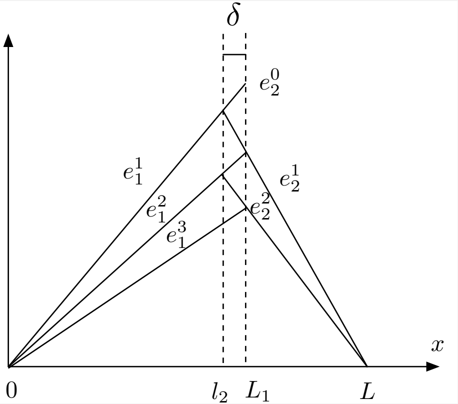
Interface Iteration

Let $\delta := L_1 - l_2$ (overlap):

$$e_2^{n+1}(L_1) = \frac{1 - \delta/(L - l_2)}{1 + \delta/l_2} \cdot e_2^n(L_1)$$

Convergence Condition

$\delta > 0$ is necessary and sufficient for convergence.



2D Fourier Analysis: Domain Setup and PDE

Domain and PDE

Decompose $\mathbb{R}^2$ into two overlapping half-planes:

- $\Omega_1 = (-\infty, \delta) \times \mathbb{R}$
- $\Omega_2 = (0, \infty) \times \mathbb{R}$

Solve:

$$(\eta - \Delta)u = f, \quad u \text{ bounded at infinity}$$

Partial Fourier Transform in y

$$\left(\eta - \frac{\partial^2}{\partial x^2} + k^2\right) \hat{e}_j^{n+1}(x, k) = 0$$

For each k, general solution:

$$\hat{e}_j^{n+1}(x, k) = \gamma_+^{n+1}(k)e^{\lambda^+(k)x} + \gamma_-^{n+1}(k)e^{\lambda^-(k)x}$$

Error Equations

Let $e_i^n := u_i^n - u|_{\Omega_i}$.

- On Ω_1: $(\eta - \Delta)e_1^{n+1} = 0$, $e_1^{n+1}(\delta, y) = e_2^n(\delta, y)$
- On Ω_2: $(\eta - \Delta)e_2^{n+1} = 0$, $e_2^{n+1}(0, y) = e_1^n(0, y)$

Boundedness Condition

To ensure boundedness:

$$\hat{e}_1^{n+1}(x, k) = \gamma_+^{n+1}(k)e^{\lambda(k)x}, \quad x < \delta$$
$$\hat{e}_2^{n+1}(x, k) = \gamma_-^{n+1}(k)e^{-\lambda(k)x}, \quad x > 0$$

with $\lambda(k) = \sqrt{\eta + k^2}$.

Convergence Factor and Iteration Behavior

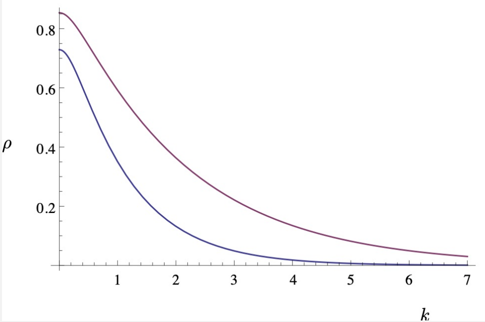
Interface Update and Factor

Matching at the interfaces gives:

$$\gamma_+^{n+1}(k) = \gamma_-^{n}(k)\, e^{-\lambda(k)\delta}$$
$$\gamma_-^{n+1}(k) = \gamma_+^{n}(k)\, e^{-\lambda(k)\delta}$$
$$\Rightarrow\ \gamma_\pm^{n+1}(k) = \rho(k)^2 \cdot \gamma_\pm^{n-1}(k)$$

Convergence Factor

$$\rho(k;\eta,\delta) = e^{-\lambda(k)\delta}, \quad \lambda(k) = \sqrt{\eta + k^2}$$



Convergence Insights

- $\Rightarrow$ **Uniform convergence:** $\rho(k) < e^{-\sqrt{\eta}\,\delta} < 1$
- $\Rightarrow$ **High-frequency error components decay rapidly**
- $\Rightarrow$ **No overlap ($\delta = 0$)** $\Rightarrow$ no decay ($\rho = 1$): stagnation

Summary of Schwarz Convergence Insights (1D and 2D)

1D Case: Iterative Affine Model

- Schwarz iterates satisfy linear error decay via interface transfer
- Convergence factor < 1 only if overlap $\delta > 0$
- No overlap $\Rightarrow$ method stagnates

2D Case: Fourier Mode Analysis

- Fourier transform yields exact decay rate per mode k
- **High-frequency modes** decay fastest
- Convergence rate: $\rho(\mathrm{k}; \eta, \delta) = \mathrm{e}^{-\sqrt{\eta+\mathrm{k}^2}\,\delta}$

Takeaway

Overlap is essential. Without it, Schwarz-type methods stall.
With it, convergence improves with frequency.

Fixed-Point and Krylov Methods: Parallel View

Fixed Point Iteration

Given a linear system $A\mathbf{x} = \mathbf{b}$, we apply an iterative fixed-point method:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + B^{-1}(\mathbf{b} - A\mathbf{x}^n)$$

This updates the guess by applying a correction based on the residual.

This can be interpreted as the fixed point of the map:

$$\mathbf{x} \mapsto \mathbf{x} + B^{-1}(\mathbf{b} - A\mathbf{x})$$

Let $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}^0$ and define $C = B^{-1}A$. Then:

$$\mathbf{x}^n = \sum_{i=0}^{n}(I - C)^i B^{-1}\mathbf{r}_0 + \mathbf{x}^0$$

Convergence condition: Spectral radius $\rho(I - C) < 1$.

Krylov Perspective

In Krylov methods, we solve the preconditioned system:

$$C\mathbf{x} = B^{-1}\mathbf{b}, \quad \text{with } C = B^{-1}A$$

Given an initial guess $\mathbf{x}^0$, the residual is:

$$\mathbf{r}^0 = B^{-1}\mathbf{b} - C\mathbf{x}^0$$

We define $\mathbf{y} := \mathbf{x} - \mathbf{x}^0$ and solve:

$$C\mathbf{y} = \mathbf{r}^0$$

Polynomial Inversion Insight

There exists a polynomial $\mathcal{P}$ of degree less than N such that:

$$C^{-1} = \mathcal{P}(C)$$

This is the foundation of Krylov subspace methods.

Krylov Methods and CG Example

Krylov Subspaces

Each iterate is constructed from a Krylov subspace:

$$\mathcal{K}^n(C, \mathbf{r}^0) := \mathrm{span}\{\mathbf{r}^0, C\mathbf{r}^0, \ldots, C^{n-1}\mathbf{r}^0\}$$

These are polynomial combinations of the initial residual, improving the approximation progressively.

Conjugate Gradient Method (SPD Case)

Objective: Given SPD matrix A, find an optimal $\mathbf{y}^n$:

$$\mathbf{y}^n = \arg \min_{\mathbf{w} \in \mathcal{K}^n(A, \mathbf{r}^0)} \|A\mathbf{w} - \mathbf{r}^0\|_{A^{-1}}$$

Then $\mathbf{x}^n = \mathbf{x}^0 + \mathbf{y}^n$.

CG Recurrence

for $i = 1, 2, \ldots$ **do**
  $\rho_{i-1} = (\mathbf{r}_{i-1}, \mathbf{r}_{i-1})$
  if $i = 1$ **then**
    $\mathbf{p}_1 = \mathbf{r}_0$
  else
    $\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$
    $\mathbf{p}_i = \mathbf{r}_{i-1} + \beta_{i-1}\mathbf{p}_{i-1}$
  end if
  $\mathbf{q}_i = A\mathbf{p}_i$
  $\alpha_i = \rho_{i-1}/(\mathbf{p}_i, \mathbf{q}_i)$
  $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\mathbf{p}_i$
  $\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i\mathbf{q}_i$
end for

GMRES and Schwarz Preconditioning

GMRES Iteration Principle

Find the best approximation in a Krylov subspace:

$$\mathbf{y}^n = \arg \min_{\mathbf{w} \in \mathcal{K}^n(C, \mathbf{r}^0)} \|C\mathbf{w} - \mathbf{r}^0\|_2$$

Preconditioned space:

$$\mathcal{K}^n(C, B^{-1}\mathbf{r}_0) = \text{span}\{B^{-1}\mathbf{r}_0, CB^{-1}\mathbf{r}_0, \ldots\}$$

Why Krylov?

Krylov methods generate optimally weighted iterates at minimal cost (especially for small n), whereas fixed-point schemes rely on static weights.

Schwarz Preconditioners

Schwarz methods serve as powerful preconditioners:

- **RAS (Restricted Additive Schwarz):** Used with GMRES or BiCGStab

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

- **ASM (Additive Schwarz Method):** Used with CG

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

Preconditioned CG with ASM: Efficiency and Convergence

Theoretical Convergence Guarantee

When using the Additive Schwarz preconditioner M_{ASM}^{-1} in Conjugate Gradient:

$$\|\mathbf{x} - \mathbf{x}_m\|_{M_{ASM}^{-1}A} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \|\mathbf{x} - \mathbf{x}_0\|_{M_{ASM}^{-1}A}$$

where $\kappa = \text{cond}(M_{ASM}^{-1}A)$ is the condition number of the preconditioned matrix.

Implication

A well-chosen M_{ASM}^{-1} significantly reduces κ, accelerating convergence of CG.

PCG Iteration with ASM Preconditioner

for $i = 1, 2, \ldots$ **do**
- $\rho_{i-1} = (\mathbf{r}_{i-1}, M_{ASM}^{-1}\mathbf{r}_{i-1})$
- **if** $i = 1$ **then**
  - $\mathbf{p}_1 = M_{ASM}^{-1}\mathbf{r}_0$
- **else**
  - $\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$
  - $\mathbf{p}_i = M_{ASM}^{-1}\mathbf{r}_{i-1} + \beta_{i-1}\mathbf{p}_{i-1}$
- **end if**
- $\mathbf{q}_i = A\mathbf{p}_i$
- $\alpha_i = \rho_{i-1}/(\mathbf{p}_i, \mathbf{q}_i)$
- $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\mathbf{p}_i$
- $\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i\mathbf{q}_i$

end for

The Krylov method applied in this case is the CG. The performance is now less sensitive to the overlap size.