Domain decomposition, hybrid methods, coarse space corrections

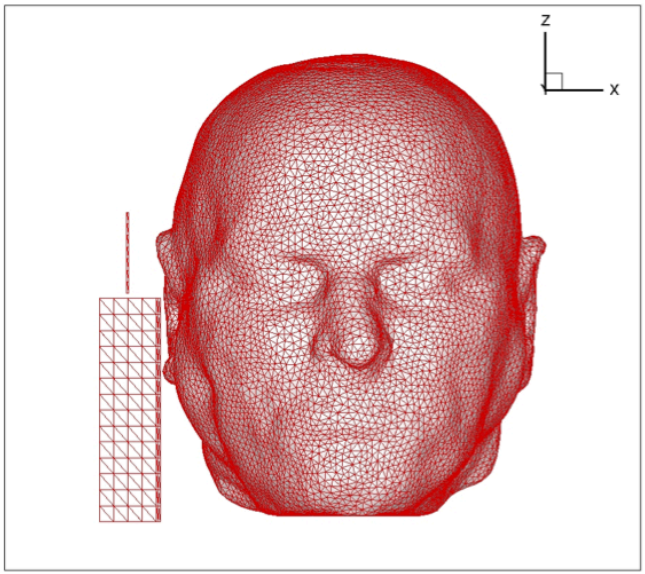
Victorita Dolean, Pierre Jolivet & Frédéric Nataf

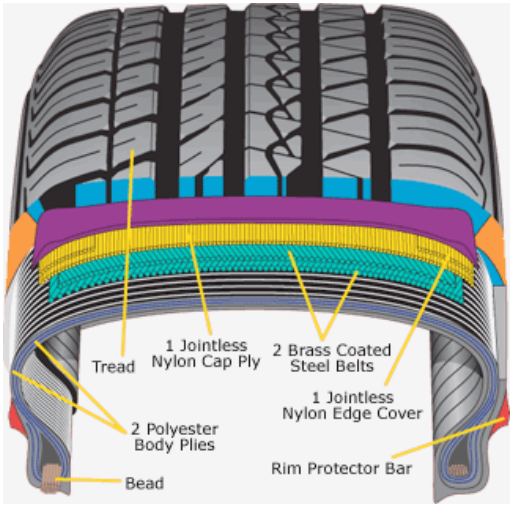
UNICE, ENSEEIHT & UPMC and CNRS

CEMRACS 2016
18 – 22 July 2016

Outline

Applications





Time-harmonic Maxwell's equations

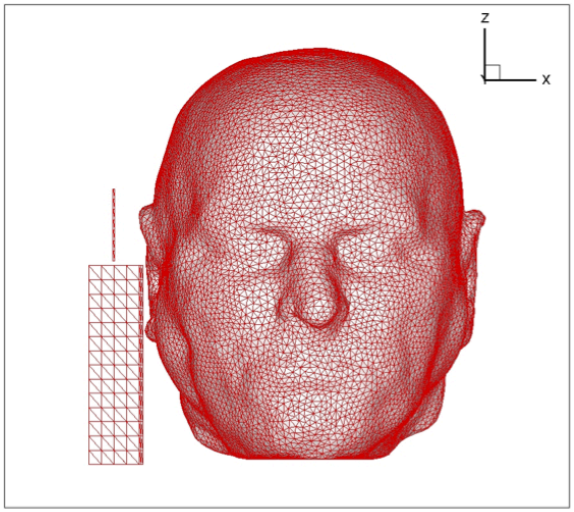
$$-i\omega\varepsilon\mathbf{E} + \nabla \times \mathbf{H} - \sigma\mathbf{E} = \mathbf{J},$$
$$i\omega\mu\mathbf{H} + \nabla \times \mathbf{E} = \mathbf{0}.$$

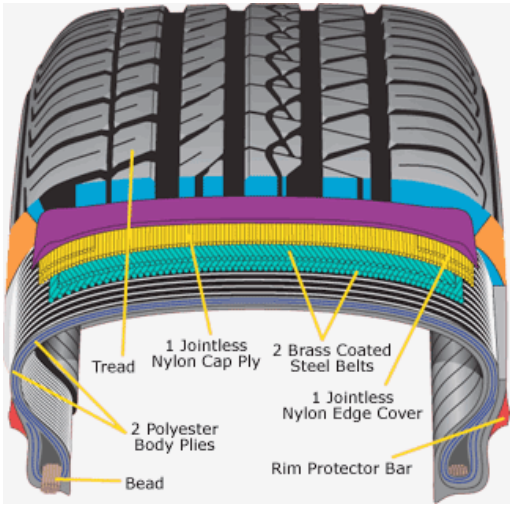
Linear elasticity

$$-\nabla \cdot (\sigma(\mathbf{u})) = \mathbf{f},$$

$$\sigma_{ij}(\mathbf{u}) = 2\mu\varepsilon_{ij}(\mathbf{u}) + \lambda\delta_{ij}\nabla \cdot (\mathbf{u}),$$
$$\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right),$$
$$\mu = \frac{E}{2(1+\nu)}, \ \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}.$$

Applications



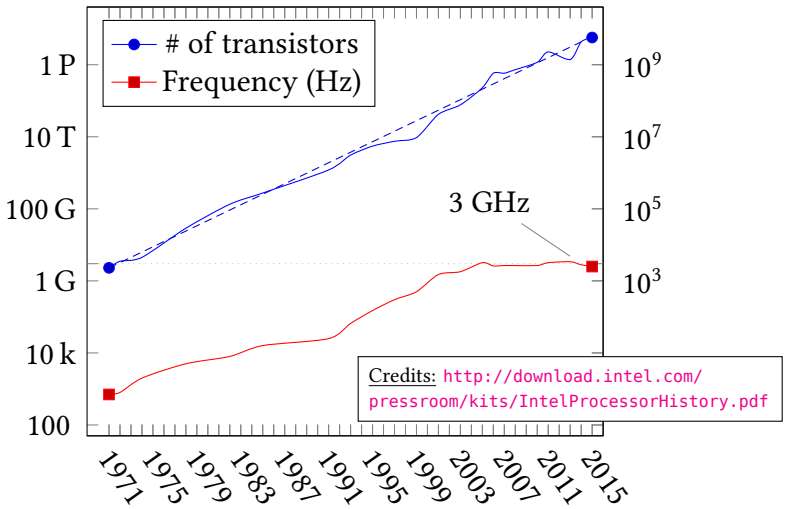


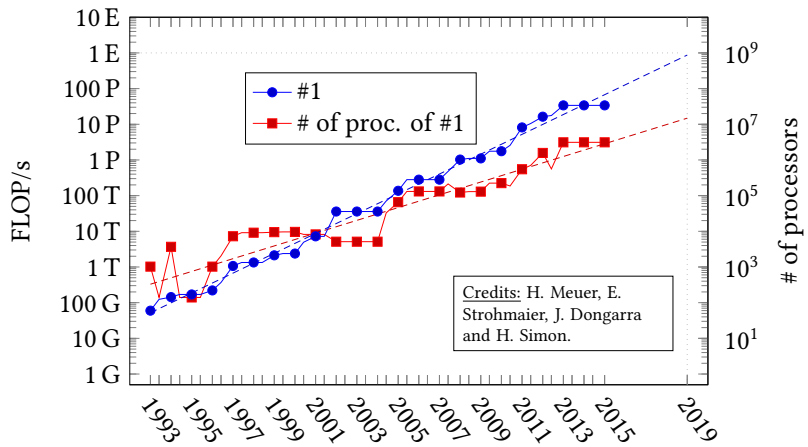
- After discretisation $\Rightarrow$ large linear system

$$A\mathbf{u} = \mathbf{b}$$

- Matrix A inherits the properties of the underlying PDE (symmetric, positive definite, indefinite, etc...)
- A is in general sparse (a lot of zeros), large (e.g. 3d applications - a few million unknowns) and ill conditioned.

Need & Opportunities for massively parallel computing





Since year 2005:

- CPU frequency stalls at 3 GHz due to the heat dissipation wall. The only way to improve the performance of computer is to go parallel
- Power consumption is an issue:
  - Large machines (hundreds of thousands of cores) cost 10-15% of their price in energy every year.
  - Smartphone, tablets, laptops (quad - octo cores) have limited power supplies

Need & Opportunities for massively parallel computing

Parallel computers are more and more available to scientists and engineers

- Apple, Linux and Windows laptops, 2/4 cores
- Desktop Computers, 6/12 cores
- Laboratory cluster, 300 cores
- University cluster, $\sim$ 2000 cores
- Cloud computing on Data Mining machines
- Supercomputers with more hundreds thousands of cores via academic (CNRS, GENCI, IDRIS, PRACE, . . .) or commercial (BULL, HP, IBM, . . .) providers

All fields of computer science are impacted.

Where to make effort in scientific computing

to compute right

- in the past: Numerical analysis of discretization schemes, a posteriori error estimates, mesh generation, reduced basis method
- Now: business as usual

to compute faster

- in the past: invest in a new machine every three years
- Now: invest every five years and add an investment in algorithmic research:

to use less energy

- in the past: nobody cared
- Now: communication avoiding algorithms , see J. Demmel, L. Grigori, M. Hoemmen, J. Langou, M. Baboulin, . . .

Need for Communication Avoiding Algorithms CAA

A simplified view of modern architectures

- Unlimited number of fast cores
- Distributed data
- Limited amount of slow and energy intensive communication

Coarse Grain algorithm

- Maximize local computations
- Minimize communications (saves time and energy altogether)
- No sequential task

$Au = f$? Panorama of linear solvers

Direct Solvers

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

Iterative Methods

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

"Hybrid Methods"

- Multigrid (A. Brandt, Ruge-Stüben, Falgout, McCormick, A. Ruhe, Y. Notay, . . .)
- Domain decomposition methods (O. Widlund, C. Farhat, J. Mandel, P.L. Lions,) are a naturally parallel compromise

Motivation: pro and cons of direct solvers

Complexity of the Gauss factorization

Gauss	$d = 1$	$d = 2$	$d = 3$
dense matrix	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
using band structure	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^{7/3})$
using sparsity	$\mathcal{O}(n)$	$\mathcal{O}(n^{3/2})$	$\mathcal{O}(n^2)$

Different sparse direct solvers

- PARDISO (http://www.pardiso-project.org)
- SUPERLU (http://crd.lbl.gov/~xiaoye/SuperLU)
- SPOOLES (www.netlib.org/linalg/spooles/spooles.2.2.html)
- MUMPS (http://graal.ens-lyon.fr/MUMPS/)
- UMFPACK (http://www.cise.ufl.edu/research/sparse/umfpack)

Why iterative solvers?

Limitations of direct solvers

In practice all direct solvers work well until a certain barrier:

- **two-dimensional problems** (10^6 unknowns)
- **three-dimensional problems** (10^5 unknowns).

Beyond, the factorization cannot be stored in memory any more.

To summarize:

- below a certain size, direct solvers are chosen.
- beyond the critical size, iterative solvers are needed.

Why domain decomposition?

Natural iterative/direct trade-off

- Parallel processing is the only way to have faster codes, new generation processors are parallel: dual, quadri core.
- Large scale computations need for an ”artificial” decomposition
- Memory requirements, direct solvers are too costly.
- Iterative solvers are not robust enough.

New iterative/direct solvers are welcome : these are domain decomposition methods

In some situations, the decomposition is natural

- Moving domains (rotor and stator in an electric motor)
- Strongly heterogeneous media
- Different physics in different subdomains

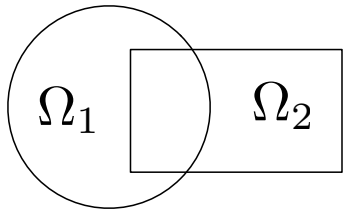
Linear Algebra from the End User point of view

Direct	DDM	Iterative
Cons: Memory Difficult to ∥ Pros: Robustness	Pro: Flexible Naurally ∥	Pros: Memory Easy to ∥ Cons: Robustness
solve(MAT,RHS,SOL)	Few black box routines Few implementations of efficient DDM	solve(MAT,RHS,SOL)

Multigrid methods: very efficient but may lack robustness, not always applicable (Helmholtz type problems, complex systems) and difficult to parallelize.

Outline

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



$$-\Delta(u) = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial\Omega.$$

Schwarz Method : $(u_1^n, u_2^n) \to (u_1^{n+1}, u_2^{n+1})$ with

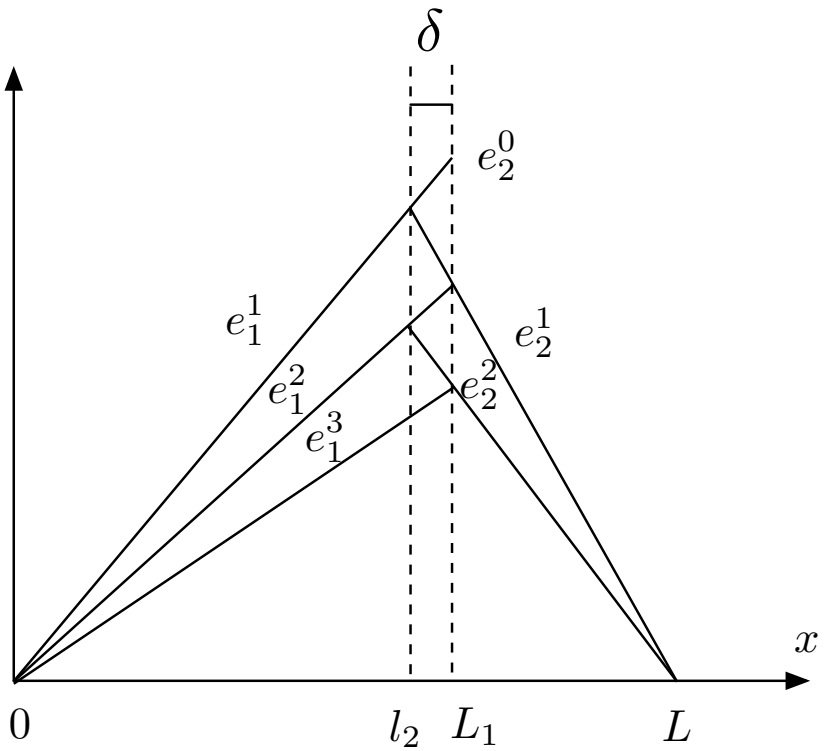
$$-\Delta(u_1^{n+1}) = f \quad \text{in } \Omega_1$$
$$u_1^{n+1} = 0 \text{ on } \partial\Omega_1 \cap \partial\Omega$$
$$u_1^{n+1} = u_2^n \quad \text{on } \partial\Omega_1 \cap \overline{\Omega_2}.$$

$$-\Delta(u_2^{n+1}) = f \quad \text{in } \Omega_2$$
$$u_2^{n+1} = 0 \text{ on } \partial\Omega_2 \cap \partial\Omega$$
$$u_2^{n+1} = u_1^{n+1} \quad \text{on } \partial\Omega_2 \cap \overline{\Omega_1}.$$

Parallel algorithm, converges but very slowly, overlapping subdomains only.
The parallel version is called **Jacobi Schwarz method (JSM)**.

Convergence in 1D

Overlap width $\delta > 0$ is sufficient and necessary to have convergence.



Fourier analysis in 2d - I

Let $\mathbb{R}^2$ decomposed into two half-planes $\Omega_1 = (-\infty, \delta) \times \mathbb{R}$ and $\Omega_2 = (0, \infty) \times \mathbb{R}$ with an overlap of size $\delta > 0$ and the problem

$$\begin{aligned} (\eta - \Delta)(u) &= f \quad \text{in} \quad \mathbb{R}^2, \\ u &\quad \text{is bounded at infinity}, \end{aligned}$$

By linearity, the errors $e_i^n := u_i^n - u|_{\Omega_i}$ satisfy the JSM $f = 0$:

$$\begin{aligned} (\eta - \Delta)(e_1^{n+1}) &= 0 \quad \text{in} \quad \Omega_1 \\ e_1^{n+1} &\quad \text{is bounded at infinity} \\ e_1^{n+1}(\delta, y) &= e_2^n(\delta, y), \end{aligned} \tag{1}$$

$$\begin{aligned} (\eta - \Delta)(e_2^{n+1}) &= 0 \quad \text{in} \quad \Omega_2 \\ e_2^{n+1} &\quad \text{is bounded at infinity} \\ e_2^{n+1}(0, y) &= e_1^n(0, y). \end{aligned} \tag{2}$$

Fourier analysis in 2d - II

By taking the partial Fourier transform of the equation in the y direction we get:

$$\left(\eta - \frac{\partial^2}{\partial x^2} + k^2\right)(\hat{e}_1^{n+1}(x,k)) = 0 \quad \text{in} \quad \Omega_1.$$

For a given k, the solution

$$\hat{e}_1^{n+1}(x,k) = \gamma_+^{n+1}(k)\exp(\lambda^+(k)x) + \gamma_-^{n+1}(k)\exp(\lambda^-(k)x).$$

must be bounded at $x = -\infty$. This implies

$$\hat{e}_1^{n+1}(x,k) = \gamma_+^{n+1}(k)\exp(\lambda^+(k)x)$$

and similarly,

$$\hat{e}_2^{n+1}(x,k) = \gamma_-^{n+1}(k)\exp(\lambda^-(k)x)$$

Fourier analysis in 2d - III

From the interface conditions we get

$$\gamma_+^{n+1}(k) = \gamma_-^n(k)\ \exp(\lambda^-(k)\delta),\ \gamma_-^{n+1}(k) = \gamma_+^n(k)\exp(-\lambda^+(k)\delta).$$

Combining these two and denoting $\lambda(k) = \lambda^+(k) = -\lambda^-(k)$, we get for $i = 1, 2$,

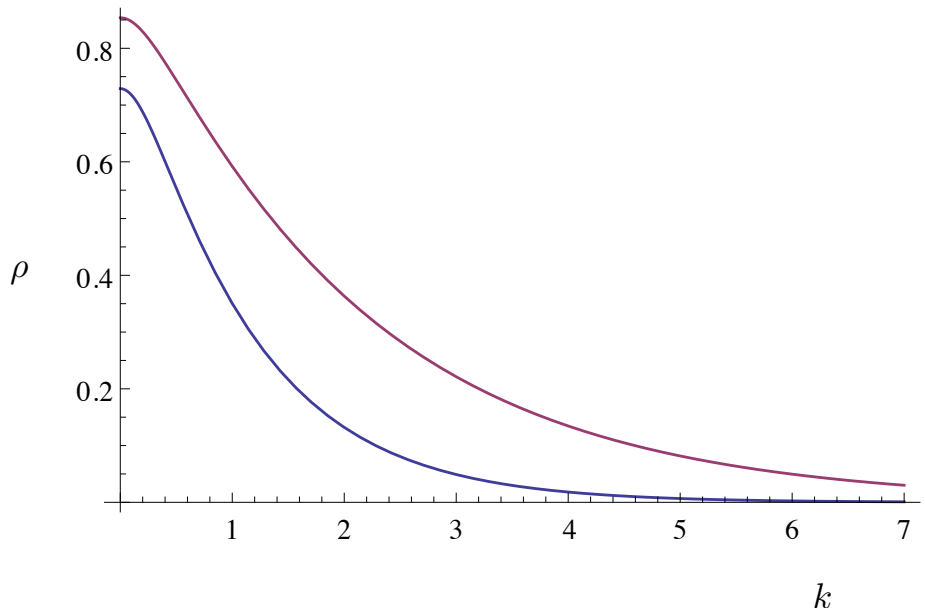
$$\gamma_\pm^{n+1}(k) = \rho(k;\alpha,\delta)^2\,\gamma_\pm^{n-1}(k)$$

with ρ the convergence rate given by:

$$\rho(k;\alpha,\delta) = \exp(-\lambda(k)\delta), \tag{3}$$

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

Fourier analysis in 2d - IV



Remark

- *For all $k \in \mathbb{R}$, $\rho(k) < \exp(-\sqrt{\eta}\,\delta) < 1$ so that $\gamma_i^n(k) \to 0$ uniformly as n goes to infinity.*
- *$\rho \to 0$ as k tends to infinity, high frequency modes of the error converge very fast.*
- *When there is no overlap ($\delta = 0$), $\rho = 1$ and there is stagnation of the method.*

Consider the discretized Poisson problem: $Au = f \in \mathbb{R}^n$.

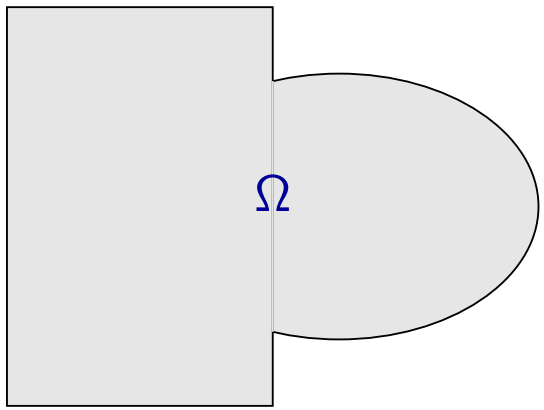
Given a decomposition of $[\![1; n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define:

- 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 $\mathbb{R}^{[\![1;n]\!]}$.

$u^m \longrightarrow u^{m+1}$ by solving concurrently:

$$u_1^{m+1} = u_1^m + A_1^{-1} R_1 (f - Au^m) \qquad u_2^{m+1} = u_2^m + A_2^{-1} R_2 (f - Au^m)$$

where $u_i^m = R_i u^m$ and $A_i := R_i A R_i^T$.



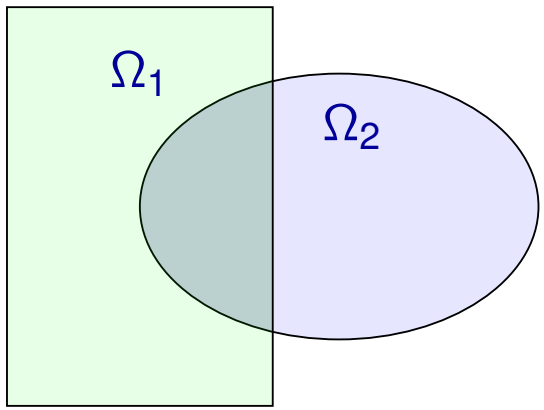
Consider the discretized Poisson problem: $Au = f \in \mathbb{R}^n$.
Given a decomposition of $[\![1; n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define:

- 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 $\mathbb{R}^{[\![1;n]\!]}$.

$u^m \longrightarrow u^{m+1}$ by solving concurrently:

$$u_1^{m+1} = u_1^m + A_1^{-1} R_1 (f - Au^m) \qquad u_2^{m+1} = u_2^m + A_2^{-1} R_2 (f - Au^m)$$

where $u_i^m = R_i u^m$ and $A_i := R_i A R_i^T$.



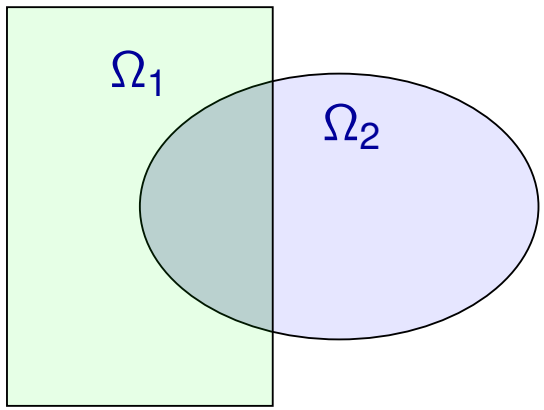
Consider the discretized Poisson problem: $Au = f \in \mathbb{R}^n$.
Given a decomposition of $[\![1; n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define:

- 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 $\mathbb{R}^{[\![1;n]\!]}$.

$u^m \longrightarrow u^{m+1}$ by solving concurrently:

$$u_1^{m+1} = u_1^m + A_1^{-1} R_1 (f - Au^m) \qquad u_2^{m+1} = u_2^m + A_2^{-1} R_2 (f - Au^m)$$

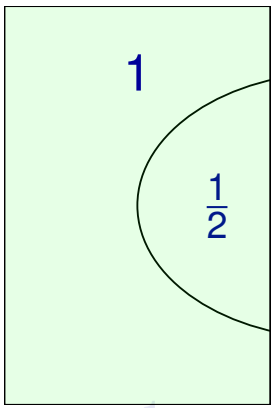
where $u_i^m = R_i u^m$ and $A_i := R_i A R_i^T$.

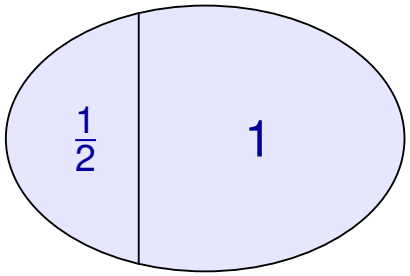


We have effectively divided, but we have yet to conquer.

Duplicated unknowns coupled via a partition of unity:

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





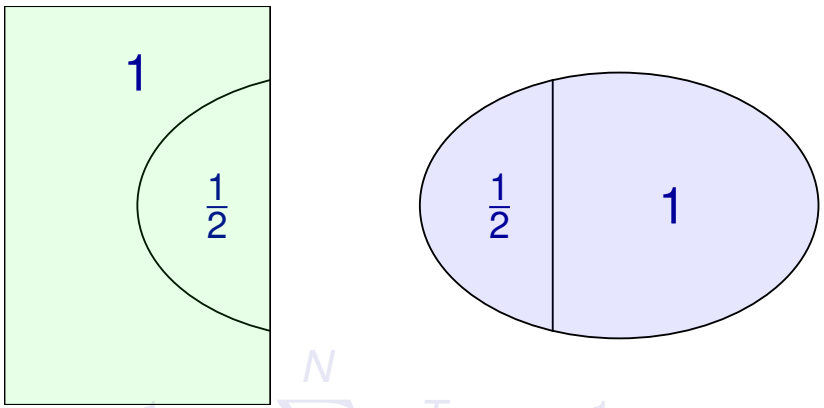
Then, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$. $M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i$.

RAS algorithm (Cai & Sarkis, 1999). Weighted Overlapping Block Jacobi method

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, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$.

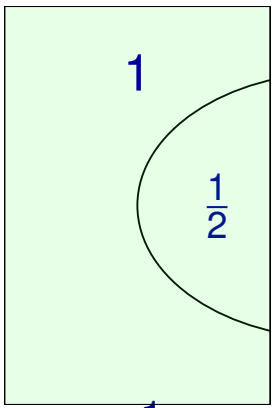
$$M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i.$$

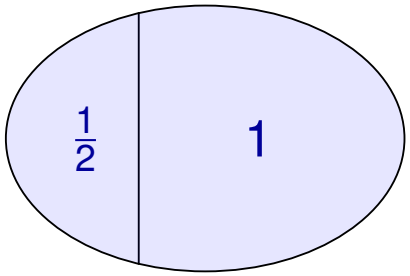
RAS algorithm (Cai & Sarkis, 1999). Weighted Overlapping Block Jacobi method

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, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$.

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

RAS algorithm (Cai & Sarkis, 1999). Weighted Overlapping Block Jacobi method

Algebraic formulation - RAS and ASM

Discrete Schwarz algorithm iterates on a pair of local functions (u_m^1, u_m^2)
RAS algorithm iterates on the global function u^m

Schwarz and RAS

Discretization of the classical Schwarz algorithm and the iterative RAS algorithm:

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

are equivalent

$$U^n = R_1^T D_1 U_1^n + R_2^T D_2 U_2^n \, .$$

(Efstathiou and Gander, 2002).

Operator M_{RAS}^{-1} is used as a preconditioner in Krylov methods for non symmetric problems.

Algebraic formulation - RAS and ASM

Discrete Schwarz algorithm iterates on a pair of local functions (u_m^1, u_m^2)
RAS algorithm iterates on the global function u^m

Schwarz and RAS

Discretization of the classical Schwarz algorithm and the iterative RAS algorithm:

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

are equivalent

$$U^n = R_1^T D_1 U_1^n + R_2^T D_2 U_2^n \, .$$

(Efstathiou and Gander, 2002).

Operator M_{RAS}^{-1} is used as a preconditioner in Krylov methods for non symmetric problems.

ASM: a symmetrized version of RAS

$$M_{RAS}^{-1} := \sum_{i=1}^{N} R_i^T \, D_i \, A_i^{-1} \, R_i \, . \tag{4}$$

A symmetrized version: Additive Schwarz Method (ASM),

$$M_{ASM}^{-1} := \sum_{i=1}^{N} R_i^T \, A_i^{-1} \, R_i \tag{5}$$

are used as a preconditioner for the conjugate gradient (CG) method. Later on, we introduce

$$M_{SORAS}^{-1} := \sum_{i=1}^{N} R_i^T \, D_i \, B_i^{-1} \, D_i \, R_i \tag{6}$$

where $(B_i)_{1 \leq i \leq N}$ are some local invertible matrices.
Although RAS is more efficient, ASM is amenable to theory.

Numerics on a toy problem

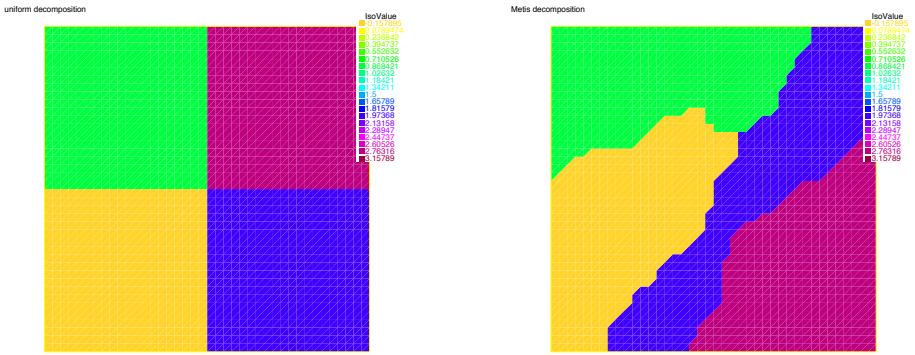


Figure: Uniform and Metis initial partitions

Overalps are added layer after layer

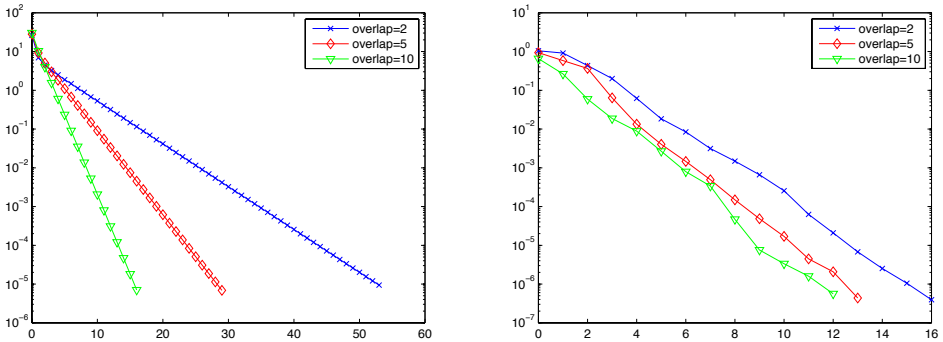


Figure: Schwarz convergence as a solver (left) and as a preconditioner (right) for different overlaps

Many cores : Strong and Weak scalability
How to evaluate the efficiency of a domain decomposition?

Strong scalability (Amdahl)

"How the solution time varies with the number of processors for a fixed *total* problem size"

Weak scalability (Gustafson)

"How the solution time varies with the number of processors for a fixed problem size *per processor*."

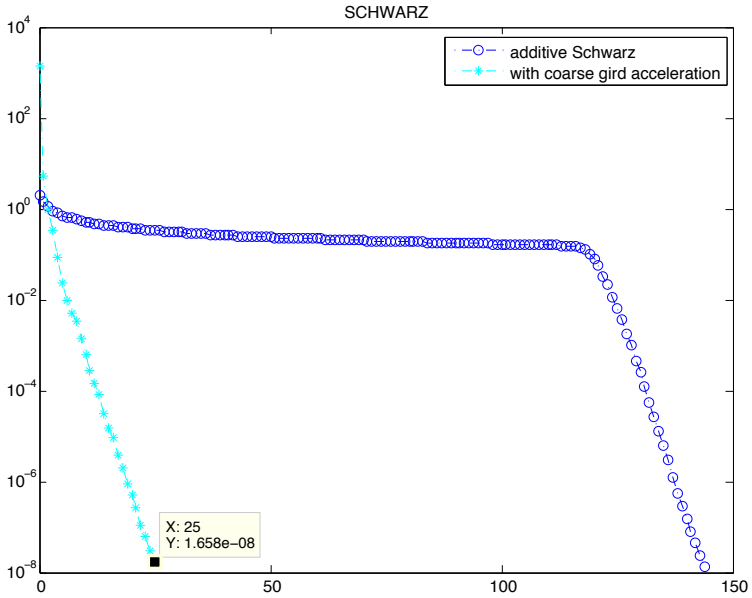
Not achieved with the one level method

Number of subdomains	8	16	32	64
ASM	18	35	66	128

The iteration number increases linearly with the number of subdomains in one direction.

Convergence curves- more subdomains

Plateaus appear in the convergence of the Krylov methods.



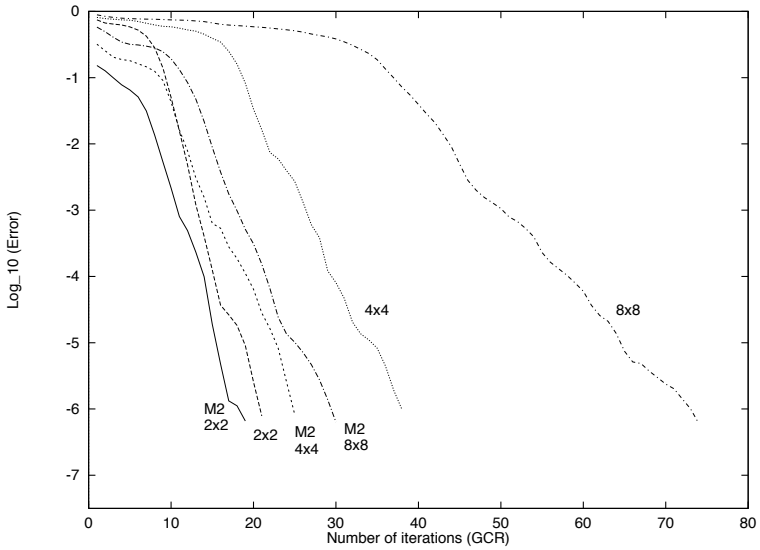


Figure: Decomposition into 64 subdomains and into $m \times m$ squares

Condition number estimate

Lemma

If there exist the constants C_1 and C_2 such that

$$C_1(M_{AS}\mathbf{x},\mathbf{x}) \leq (A\mathbf{x},\mathbf{x}) \leq C_2(M_{AS}\mathbf{x},\mathbf{x}),\ \forall\,\mathbf{x}\in\mathbb{R}^n \tag{7}$$

then $\lambda_{max}(M_{AS}^{-1}A) \leq C_2$, $\lambda_{min}(M_{AS}^{-1}A) \geq C_1$ and thus $\kappa(M_{AS}^{-1}A) \leq C_2/C_1$.

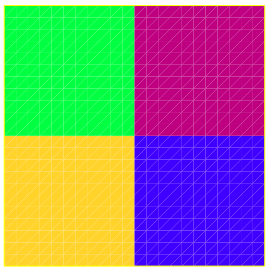
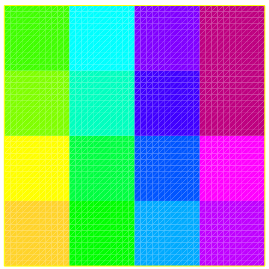
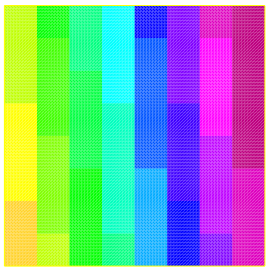
$\kappa(M_{AS}^{-1}A)$ independent of N (number of subdomains) $\Rightarrow$ the execution time will be independent of the number of processors.

Lemma

Let $col(j) \in \{1,\ldots,\mathcal{N}^c\}$ be the color of the domain j defined such that $(AR_k^T\mathbf{x}_k, R_l^T\mathbf{x}_l) = 0$ if $col(k) = col(l)$. Then $\lambda_{max}(M_{AS}^{-1}A) \leq \mathcal{N}_c$.

Why the algorithm is not scalable?

We have that $\lambda_{max}(M_{AS}^{-1}A) \leq \mathcal{N}_c << N$ (usual decomposition) BUT $\lambda_{min}(M_{AS}^{-1}A)$ depends on N.
Numerical experiment: subdomain = square with 20×20 discretisation points with two layers of overlap.

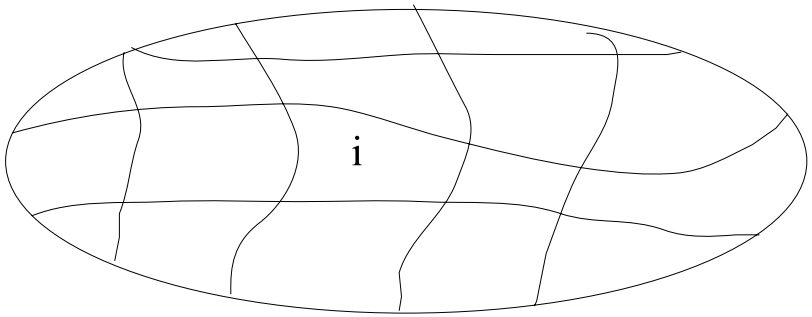
Solution of a Poisson problem $-\Delta u = f$

Number of subdomains	2x2	4x4	8x8
Number of iterations	20	36	64

How to achieve scalability

Stagnation corresponds to a few very low eigenvalues in the spectrum of the preconditioned problem. They are due to the lack of a global exchange of information in the preconditioner.

$$-\Delta u = f \text{ in } \Omega$$
$$u = 0 \text{ on } \partial\Omega$$



The mean value of the solution in domain i depends on the value of f on all subdomains.
A classical remedy consists in the introduction of a coarse problem that couples all subdomains. This is closely related to deflation technique classical in linear algebra (see Y. Saad, J. Erhel, Nabben and Vuik) and multigrid techniques.

Adding a coarse space

One level methods are not scalable for steady state problems.

We add a coarse space correction (*aka* second level)
Let V_H be the coarse space and Z be a basis, $V_H = \operatorname{span} Z$, writing $R_0 = Z^T$ we define the two level preconditioner as:

$$M^{-1}_{ASM,2} := R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i.$$

The Nicolaides approach (1987) is to use the kernel of the operator as a coarse space, this is the constant vectors, in local form this writes:

$$Z := (R_i^T D_i R_i \mathbf{1})_{1 \leq i \leq N}$$

where D_i are chosen so that we have a partition of unity:

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

Theoretical convergence result

Theorem (Widlund, Dryja)

Let $M_{ASM,2}^{-1}$ *be the two-level additive Schwarz method:*

$$\kappa(M_{ASM,2}^{-1} A) \leq C \left(1 + \frac{H}{\delta}\right)$$

where δ *is the size of the overlap between the subdomains and* H *the subdomain size.*

This does indeed work very well

Number of subdomains	8	16	32	64
ASM	18	35	66	128
ASM + Nicolaides	20	27	28	27

Other Deflation and Coarse grid correction

Let A be a SPD matrix, we want to solve

$$Ax = b$$

with a preconditioner M (for example the Schwarz method).
Let Z be a rectangular matrix so that the "bad eigenvectors" belong to the space spanned by its columns. Define

$$P := I - AQ, \quad Q := ZE^{-1}Z^T, \quad E := Z^T AZ,$$

Additive correction formulas:

$$\mathcal{P}_{A-add} := M^{-1} + Q \qquad \text{(Additive, Nicolaides, 1987)}$$

$$\mathcal{P}_{BNN} := P^T M^{-1} P + Q \quad \text{(Balanced, Mandel, 1993)}$$

$$\mathcal{P}_{A-DEF2} := P^T M^{-1} + Q, \quad \text{(Deflated, Vuik et al., 20xx)}$$

Let r_n be the residual at step n of the algorithm, for any Krylov method: $Z^T r_n = 0$ provided $Z^T r_0 = 0$.

Outline

P.L. Lions' Algorithm (1988)

$$
\begin{aligned}
&-\Delta(u_1^{n+1}) = f \quad \text{in } \Omega_1, \\
&u_1^{n+1} = 0 \quad \text{on } \partial\Omega_1 \cap \partial\Omega, \\
&(\frac{\partial}{\partial n_1} + \alpha)(u_1^{n+1}) = (-\frac{\partial}{\partial n_2} + \alpha)(u_2^n) \quad \text{on } \partial\Omega_1 \cap \overline{\Omega_2},
\end{aligned}
$$

(n_1 and n_2 are the outward normal on the boundary of the subdomains)

$$
\begin{aligned}
&-\Delta(u_2^{n+1}) = f \quad \text{in } \Omega_2, \\
&u_2^{n+1} = 0 \quad \text{on } \partial\Omega_2 \cap \partial\Omega \\
&(\frac{\partial}{\partial n_2} + \alpha)(u_2^{n+1}) = (-\frac{\partial}{\partial n_1} + \alpha)(u_1^n) \quad \text{on } \partial\Omega_2 \cap \overline{\Omega_1}.
\end{aligned}
$$

with $\alpha > 0$. Overlap is not necessary for convergence.
Parameter α can be optimized for.
Extended to the Helmholtz equation (B. Després, 1991)
a.k.a FETI 2 LM (Two-Lagrange Multiplier Method), 1998.

A model problem

$$\mathcal{L}(u) := \eta u - \Delta u = f \text{ in } \mathbb{R}^2, \quad \eta > 0$$

The plane $\mathbb{R}^2$ is divided into two half-planes with an overlap of size $\delta \geq 0$ and the algorithm writes:

$$\begin{aligned}
&\mathcal{L}(u_1^{n+1}) = f \quad \text{in } \Omega_1 :=]-\infty, \delta[\times \mathbb{R}, \\
&(\frac{\partial}{\partial n_1} + \alpha)(u_1^{n+1}) = (-\frac{\partial}{\partial n_2} + \alpha)(u_2^n) \quad \text{at } x = \delta \\
&\mathcal{L}(u_2^{n+1}) = f \quad \text{in } \Omega_2 :=]0, \infty[\times \mathbb{R}, \\
&(\frac{\partial}{\partial n_2} + \alpha)(u_2^{n+1}) = (-\frac{\partial}{\partial n_1} + \alpha)(u_1^n) \quad \text{at } x = 0
\end{aligned}$$

A Fourier analysis leads to the following convergence rate (k is the dual variable):

$$\rho(k; \delta, \alpha) = \left| \frac{\sqrt{\eta + k^2} - \alpha}{\sqrt{\eta + k^2} + \alpha} \right| e^{-\sqrt{\eta + k^2}\,\delta}$$

Overlapping Subdomains

Overlapping subdomains – Implementation issues

A direct discretization of the P.L. Lions algorithm is doable but not easy:

- the right hand side has to be computed in the interior of the subdomain
- it involves normal derivatives to the interfaces

Fix ORAS preconditioner

Let B_i be the matrix of the Robin subproblem in each subdomain $1 \leq i \leq N$, define

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

Optimized multiplicative, additive, and restricted additive Schwarz preconditioning, St Cyr, M. Gander et al, 2007

P.L. Lions and ORAS

Provided subdomains overlap, discretization of the classical P.L. Lions algorithm and the iterative ORAS algorithm:

$$U^{n+1} = U^n + M_{ORAS}^{-1} r^n \, , r^n := F - A\,U^n.$$

are equivalent

$$U^n = R_1^T D_1 U_1^n + R_2^T D_2 U_2^n \, ,$$

(St Cyr, Gander and Thomas, 2007).

- **Huge** simplification in the implementation: no boundary right hand side discretization
- Operator M_{ORAS}^{-1} is used as a preconditioner in Krylov methods for non symmetric problems.
- *First step in a global theory*

Helmholtz Equation

We want to solve

$$\begin{aligned} -\omega^2 u - \Delta u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

Schwarz method is problematic:
Subproblems may be ill posed if ω^2 is close to an eigenvalue of the Laplace operator with Dirichlet conditions.

Fourier analysis
The convergence rate of the classical Schwarz method is:

$$\rho = e^{-\sqrt{-\omega^2 + k^2}\,\delta}$$

No damping for propagative modes $\Longrightarrow$ very bad convergence

B. Després' Algorithm, 1991

$$
\begin{aligned}
&-\omega^2 u_1^{n+1} - \Delta(u_1^{n+1}) = f \quad \text{in } \Omega_1, \\
&(\frac{\partial}{\partial n_1} + I\omega)(u_1^{n+1}) = (-\frac{\partial}{\partial n_2} + I\omega)(u_2^n) \quad \text{on } \partial\Omega_1 \cap \overline{\Omega_2},
\end{aligned}
$$

(n_1 and n_2 are the outward normal on the boundary of the subdomains)

$$
\begin{aligned}
&-\Delta(u_2^{n+1}) = f \quad \text{in } \Omega_2, \\
&(\frac{\partial}{\partial n_2} + I\omega)(u_2^{n+1}) = (-\frac{\partial}{\partial n_1} + I\omega)(u_1^n) \quad \text{on } \partial\Omega_2 \cap \overline{\Omega_1}.
\end{aligned}
$$

Extended to the Mawell system (B. Després, 1991)
a.k.a FETI 2 LM (Two-Lagrange Multiplier Method), 1998.

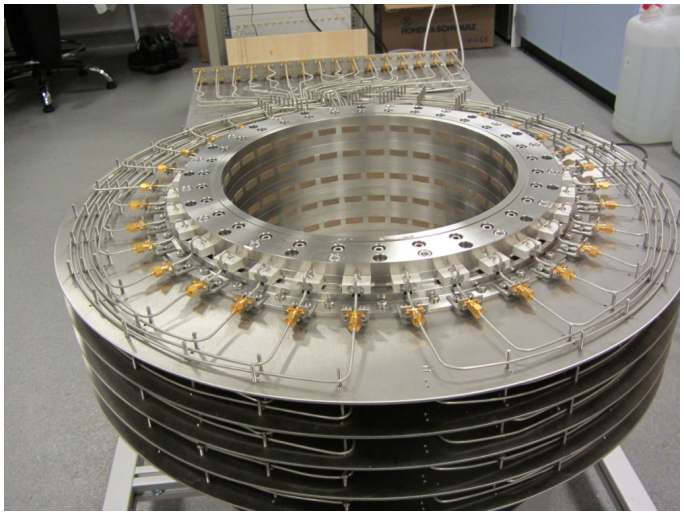
Helmholtz equation – Overlapping subdomains $\delta > 0$

It is possible to study the convergence rate in the Fourier space:

$$\rho(k) \equiv \begin{cases} \left| \dfrac{I\sqrt{\omega^2 - k^2} - I\omega}{I\sqrt{\omega^2 - k^2} + I\omega} \right| \exp^{-I\sqrt{\omega^2 - k^2}\delta} & \text{if } |k| < \omega \ (I^2 = -1) \\ \left| \dfrac{\sqrt{k^2 - \omega^2} - I\omega}{\sqrt{k^2 - \omega^2} + I\omega} \right| \exp^{-\sqrt{k^2 - \omega^2}\delta} & \text{if } |k| > \omega \end{cases}$$

Moreover, a Krylov method (GC, GMRES, BICGSTAB, ...) replaces the fixed point algorithm.

Parallel Software tools : HPDDM and FreeFem++



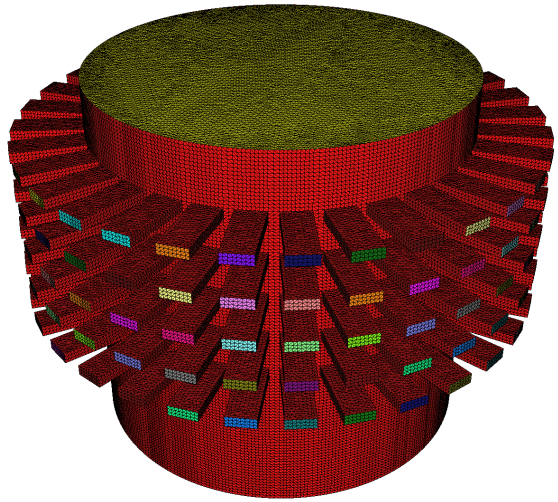


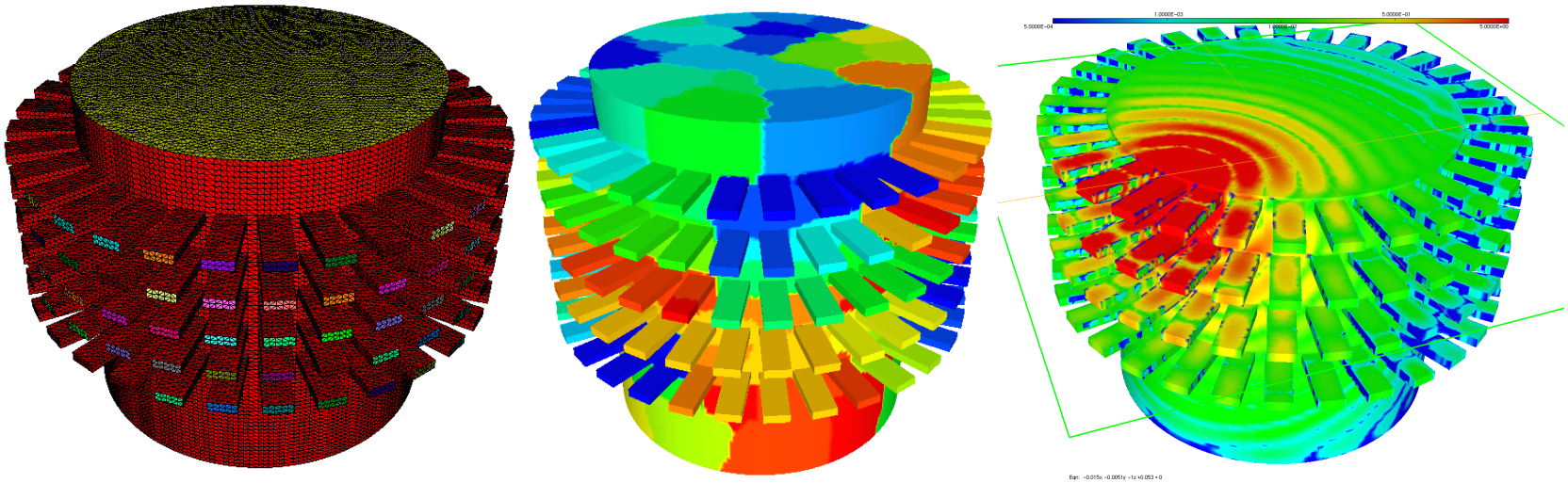
Figure: Antennas and mesh – interior diameter 28,5 cm

Two in-house open source libraries (LGPL) linked to many third-party libraries:

- HPDDM (High Performance Domain Decomposition Methods) for massively parallel computing
- FreeFem++(-mpi) for the parallel simulation of equations from physics by the finite element method (FEM).

Forward problem and Synthetic data

- Mesh with 2.3M degrees of freedom;
- Domain decomposition methods with impedance interface conditions, twice as fast as Dirichlet interface conditions;
- Parallel computing on 64 cores on SGI UV2000 at UPMC : 3s per emitter, 5 mn as a whole.



Non Overlapping Subdomains

Helmholtz Equation – Non Overlapping decomposition

M. Gander, F. Nataf, F. Magoulès
SIAM J. Sci. Comp., 2002.

We want to solve

$$-\omega^2 u - \Delta u = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial\Omega.$$

The relaxation algorithm is : $(u_1^p, u_2^p) \to (u_1^{p+1}, u_2^{p+1})$ with $(i \neq j, i = 1, 2)$

$$(-\omega^2 - \Delta)(u_i^{p+1}) = f \quad \text{in } \Omega_i$$
$$(\frac{\partial}{\partial n_i} + \mathcal{S})(u_i^{p+1}) = (-\frac{\partial}{\partial n_j} + \mathcal{S})(u_j^p) \quad \text{on } \Gamma_{ij}.$$
$$u_i^{p+1} = 0 \text{ on } \partial\Omega_i \cap \partial\Omega$$

The operator $\mathcal{S}$ has the form

$$\mathcal{S} = \alpha - \gamma \frac{\partial^2}{\partial \tau^2} \qquad \alpha, \gamma \in \mathbb{C}$$

Application: the Helmholtz Equation

By choosing carefully the coefficients α and γ, it is possible to optimize the convergence rate of the iterative method which in the Fourier space is given by

$$\rho(k;\alpha,\gamma) \equiv \begin{cases} \left| \dfrac{I\sqrt{\omega^2 - k^2} - (\alpha + \gamma k^2)}{I\sqrt{\omega^2 - k^2} + (\alpha + \gamma k^2)} \right| & \text{if } |k| < \omega \quad (I^2 = -1) \\ \left| \dfrac{\sqrt{k^2 - \omega^2} - (\alpha + \gamma k^2)}{\sqrt{k^2 - \omega^2} + (\alpha + \gamma k^2)} \right| & \text{if } |k| > \omega \end{cases}$$

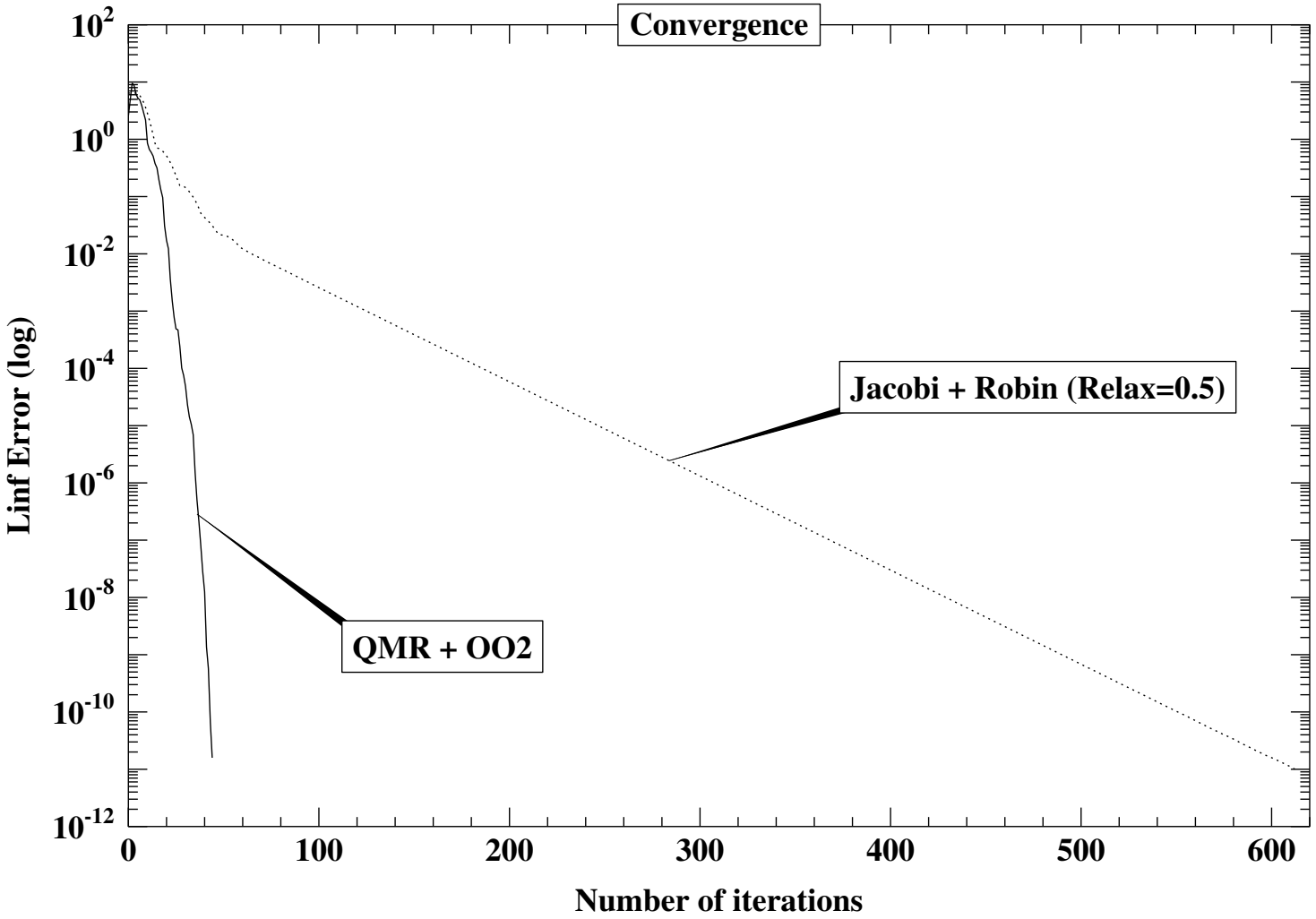
Finally, we get analytic formulas for α and γ (h is the mesh size):

$$\alpha_{opt} = \alpha(\omega, h) \text{ and } \gamma_{opt} = \gamma(\omega, h),$$

Moreover, a Krylov method (GC, GMRES, BICGSTAB, ...) replaces the fixed point algorithm.

The Helmholtz Equation – Numerical Results

Waveguide: Optimized Schwarz method with QMR compared to ABC0 ($\partial_n + I\omega$) with relaxation on the interface



Chevalier and N., 1997

Discretization of the two-field formulation

A direct discretization would require the computation of the normal derivatives along the interfaces in order to evaluate the right handsides. We introduce two new variables

$$\lambda^1 = -\frac{\partial u_2}{\partial n_2} + \mathcal{S}(u_2) \text{ and } \lambda^2 = -\frac{\partial u_1}{\partial n_1} + \mathcal{S}(u_1).$$

The algorithm reads now

$$\begin{aligned}
-\Delta u_1^{n+1} + \omega^2 u_1^{n+1} &= f \text{ in } \Omega_1 \\
\frac{\partial u_1^{n+1}}{\partial n_1} + \mathcal{S}(u_1^{n+1}) &= {\lambda^1}^n \text{ on } \Gamma_{12} \\
-\Delta u_2^{n+1} + \omega^2 u_2^{n+1} &= f \text{ in } \Omega_2 \\
\frac{\partial u_2^{n+1}}{\partial n_2} + \mathcal{S}(u_2^{n+1}) &= {\lambda^2}^n \text{ on } \Gamma_{12} \\
{\lambda^1}^{n+1} &= -{\lambda^2}^n + (\mathcal{S}+\mathcal{S})(u_2^{n+1}({\lambda^1}^p, f)) \\
{\lambda^2}^{n+1} &= -{\lambda^1}^n + (\mathcal{S}+\mathcal{S})(u_1^{n+1}({\lambda^2}^p, f)).
\end{aligned}$$

Finite Element Discretization

A finite element discretization leads to the following linear system:

$$\begin{aligned} \lambda^1 &= -\lambda^2 + (S+S)B^2u^2 \\ \lambda^2 &= -\lambda^1 + (S+S)B^1u^1 \\ \widetilde{K}^1u^1 &= f^1 + {B^1}^T\lambda^1 \\ \widetilde{K}^2u^2 &= f^2 + {B^2}^T\lambda^2 \end{aligned} \tag{8}$$

where B^1 (resp. B^2) is the trace operator of domain Ω^1 (resp. Ω^2) on the interface Γ_{12}. Matrix $\widetilde{K}^i,\ i=1,2$ arises from the discretization of the local Helmholtz subproblems along with the interface condition $\partial_n + \alpha - \gamma\partial_{\tau\tau}$.

$$\widetilde{K}^i = K^i - \omega^2 M^i + {B^i}^T(\alpha M_{\Gamma_{12}} + \gamma K_{\Gamma_{12}})B^i \tag{9}$$

where K^i is the stiffness matrix, M^i the mass matrix, $M_{\Gamma_{12}}$ is the interface mass matrix and $K_{\Gamma_{12}}$ is the interface stiffness matrix.

More precisely, the interface mass matrix $M_{\Gamma_{12}}$ and the interface stiffness matrix $K_{\Gamma_{12}}$ are defined by

$$[M_{\Gamma_{12}}]_{lm} = \int_{\Gamma_{12}} \phi_l \phi_m d\xi \quad \text{and} \quad [K_{\Gamma_{12}}]_{lm} = \int_{\Gamma_{12}} \nabla_\tau \phi_l \nabla_\tau \phi_m d\xi \quad (10)$$

where ϕ_l et ϕ_m are the basis functions associated to nodes l and m on the interface Γ_{12} and $\nabla_\tau \phi$ is the tangential component of $\nabla \phi$ on the interface.
We have

$$S = \alpha M_{\Gamma_{12}} + \gamma K_{\Gamma_{12}}.$$

The substructured linear system of the two-field formulation has the form

$$F\lambda = d \tag{11}$$

where $\lambda = (\lambda^1, \lambda^2)$, F is a matrix and d is the right handside

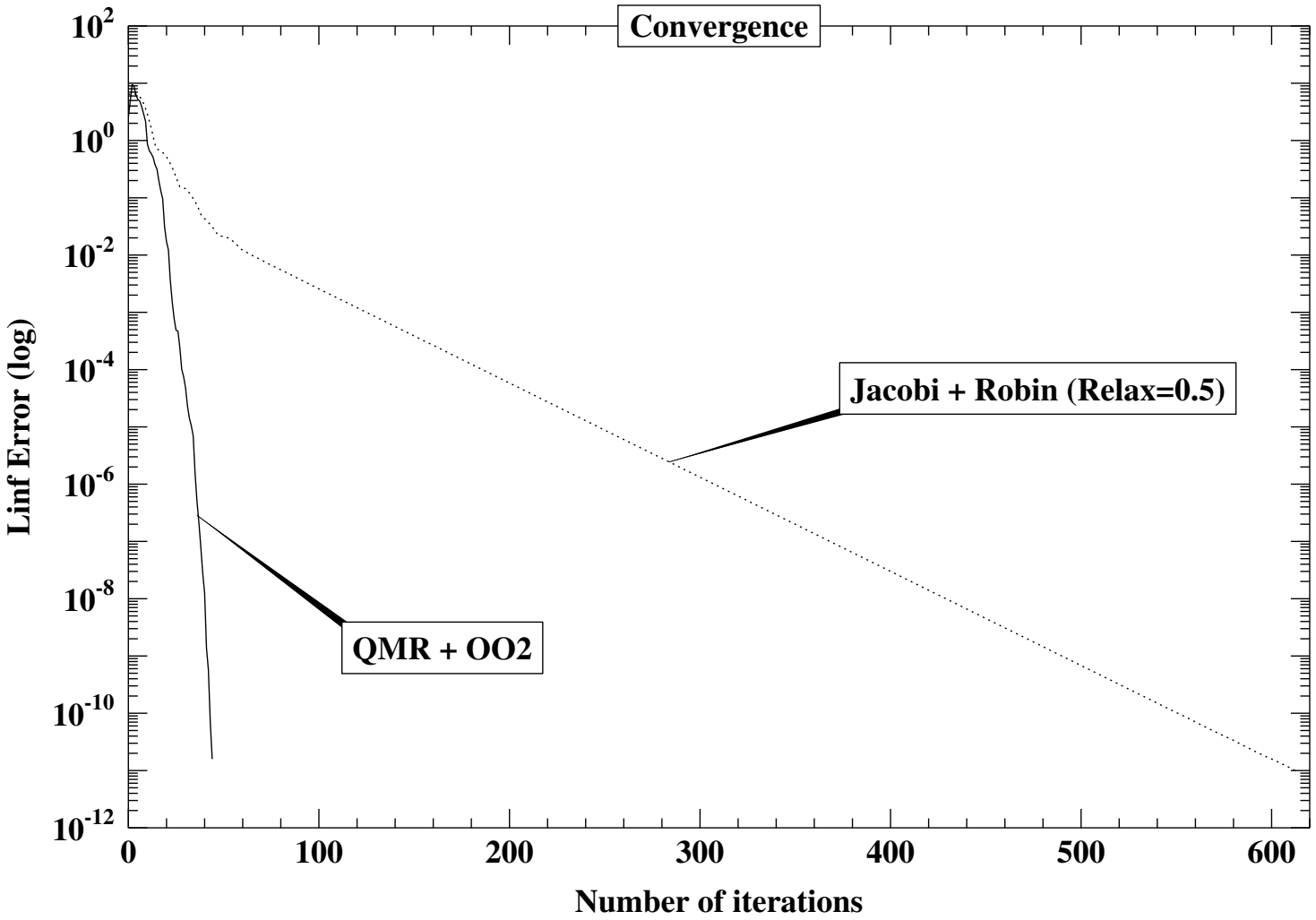
$$F = \begin{bmatrix} I & I - (S + S)B^2 \widetilde{K}^{2^{-1}} B^{2^T} \\ I - (S + S)B^1 \widetilde{K}^{1^{-1}} B^{1^T} & I \end{bmatrix}$$

$$d = \begin{bmatrix} (S + S)B^1 \widetilde{K}^{1^{-1}} f^1 \\ (S + S)B^2 \widetilde{K}^{2^{-1}} f^2 \end{bmatrix}$$

The linear system is solved by a Krylov type method, here the ORTHODIR algorithm. The matrix vector product amounts to solving a subproblem in each subdomain and to send interface data between subdomains.

General Interface Conditions for the Helmholtz Equation
Numerical Results

Waveguide: Optimized Schwarz method with QMR and ABC0 ($\partial_n + I\omega$) with relaxation on the interface



General Interface Conditions for the Helmholtz Equation
Numerical Results

Acoustic in a Car : Iteration Counts for various interface conditions

N_s	ABC 0	ABC 2	Optimized
2	16 it	16 it	9 it
4	50 it	52 it	15 it
8	83 it	93 it	25 it
16	105 it	133 it	34 it

ABC 0: Absorbing Boundary Conditions of Order 0 ($\partial_n + I\omega$)
ABC 2: Absorbing Boundary Conditions of Order 2 ($\partial_n + I\omega - 1/(2I\omega)\partial_{y^2}$)
Optimized: Optimized Interface Conditions

Maxwell equations

Other works on Maxwell's equations

Després, ; Joly, ; Roberts, A domain decomposition method for the harmonic Maxwell equations. Iterative methods in linear algebra , 1992.

Dolean, ; Gander, ; Gerardo-Giorda, Optimized Schwarz methods for Maxwell's equations. SISC, 2009

They are currently used in electromagnetic simulations:
LEE Jin-Fa - Ohio State University, ECE Department, USA:

Z. Peng, K. H. Lim, and J. F. Lee, Computations of Electromagnetic Wave Scattering from Penetrable Composite Targets using a Surface Integral Equation Method with Multiple Traces, IEEE T. ANTENNA PROPAG., 2012.

Z. Peng, K. H. Lim, and J. F. Lee, Non-conformal Domain Decomposition Methods for Solving Large Multi-scale Electromagnetic Scattering Problems, Proceeding of IEEE, 2012.

Outline

Some multigrid solvers (free and commercial)

- AmgX - NVIDIA Developer (https://developer.nvidia.com/amgx)
- AMG via HYPER (http://computation.llnl.gov/project/linear_solvers/software.php)
- PCGAMG via PETSC (http://www.mcs.anl.gov/petsc)
- AGMG (http://homepages.ulb.ac.be/~ynotay/)
- SAMG (http://www.scai.fraunhofer.de/en/business-research-areas/numerical-software/products/samg.html)

Geometric Multigrid Methods

(Seen as a special case of Domain Decomposition Methods)

One subdomain equals one cell. Additive Schwarz method reduces to the Jacobi method $\longrightarrow$ Fine level preconditioner :

$$M_{Jacobi}^{-1} := diag(diag(A))^{-1}.$$

High frequency modes of the error are quickly damped by the Jacobi (or Gauss-Seidel) method.

Coarse Space Grid Correction damps Low frequency modes, a coarser discretization is introduced. Let I_{2h}^{h} be an interpolation operator from a coarse grid ($2h$) to the fine grid (h).
Let $R_0^T := I_{2h}^{h}$ and

$$M_{MG2}^{-1}[A] := R_0^T (R_0 A R_0^T)^{-1} R_0 + M_{Jacobi}^{-1}.$$

Simple Two grid method

More elaborate corrections:

- More levels and Recursive approach
  - Apply to $(R_0 A R_0^T)$ the same strategy by introducing a third coarse grid ($4h$) and the interpolation operator $R_1^T := I_{4h}^{2h}$

$$M_{MG2}^{-1}[R_0 A R_0^T] := R_1^T (R_1 R_0 A R_0^T R_1^T)^{-1} R_1 + diag(diag(R_0 A R_0^T))^{-1}.$$

$$M_{MG3}^{-1}[A] := R_0^T M_{MG2}^{-1}[R_0 A R_0^T] R_0 + M_{Jacobi}^{-1},$$

- Various strategies to move across levels: V and W cycles

Decomposition in the frequency domain rather than in space.

Recall

$$P := I - AQ, \quad Q := ZE^{-1}Z^T, \quad E := Z^T AZ,$$

Some properties: $QAZ = Z$, $P^T Z = 0$ and $P^T Q = 0$.

$$\mathcal{P}_{A-DEF2} := P^T M^{-1}_{Jacobi} + Q,$$

$$\mathcal{P}_{BNN} := P^T M^{-1} P + Q \text{ (Mandel, 1993)}$$

Let r_n be the residual at step n of the algorithm: $Z^T r_n = 0$.

Multigrid $V(1,1)$-cycle

$$M^{-1}_{MG} := M^{-1}_{Jacobi} P + P^T M^{-1}_{Jacobi} + Q - M^{-1}_{Jacobi} P M^{-1}_{Jacobi}$$

Aggregation Multigrid Methods

When you have no access to the underlying grid, it is still possible to aggregate d.o.f's by exploiting the graph of the matrix.
Two-level preconditioner, by grouping every three d.o.f's :

$$Z := \left(\begin{array}{c|c|c} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right)$$

Let $R_0^T := Z$ and

$$M_{AMG2}^{-1}[A] := R_0^T (R_0 A R_0^T)^{-1} R_0 + M_{Jacobi}^{-1}.$$

Algebraic Multigrid Methods

Pros:

- Optimality for Poisson or Darcy problem even with highly heterogeneous coefficients
- Black box implementations
- Weakly scalable

Cons

- Difficulties and even failures with systems of PDEs
- Not so black box since it needs the near kernel of the operator
- Fails for Wave Propagation phenomena in the frequency domain (shifted Laplacian: Erlangga, Osterlee and Vuik)
- Less theory than for DDM (Notay)

Direct Solvers

Gauss or LU factorization

$$A = LU.$$

where L is a lower triangular matrix and U is an upper triangular matrix.
Different sparse direct solvers (free and commercial)

- PARDISO (http://www.pardiso-project.org)
- SUPERLU (http://crd.lbl.gov/~xiaoye/SuperLU)
- SPOOLES (www.netlib.org/linalg/spooles/spooles.2.2.html)
- MUMPS (http://graal.ens-lyon.fr/MUMPS/)
- UMFPACK (http://www.cise.ufl.edu/research/sparse/umfpack)

Multifrontal : a way to break sequentiality

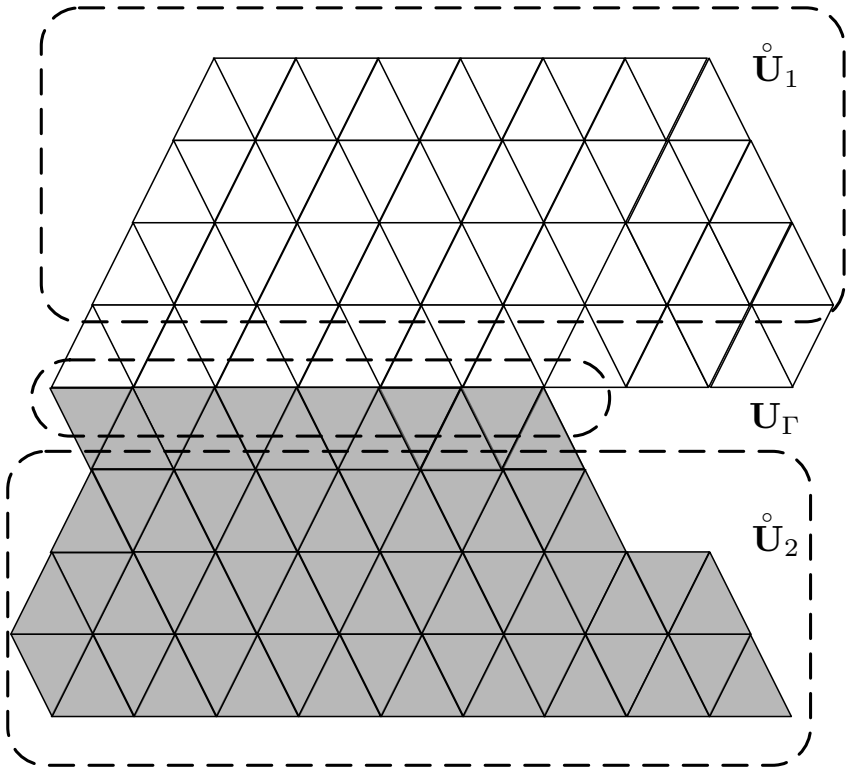


Figure: Degrees of freedom partition for a multifrontal method

Multifrontal factorizations

By numbering interface equations last, this leads to a block decomposition of the linear system which has the shape of an arrow (pointing down to the right):

$$\begin{pmatrix} A_{11} & 0 & A_{1\Gamma} \\ 0 & A_{22} & A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathring{\mathbf{U}}_1 \\ \mathring{\mathbf{U}}_2 \\ \mathbf{U}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathring{\mathbf{F}}_1 \\ \mathring{\mathbf{F}}_2 \\ \mathbf{F}_\Gamma \end{pmatrix}. \tag{12}$$

A simple computation shows that we have a block factorization of matrix A

$$A = \begin{pmatrix} I & & \\ 0 & I & \\ A_{\Gamma 1}A_{11}^{-1} & A_{\Gamma 2}A_{22}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & & \\ & A_{22} & \\ & & \mathbb{S} \end{pmatrix} \begin{pmatrix} I & 0 & A_{11}^{-1}A_{1\Gamma} \\ & I & A_{22}^{-1}A_{2\Gamma} \\ & & I \end{pmatrix}.$$

$$\boxed{\mathbb{S} := A_{\Gamma\Gamma} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma}}$$

is a Schur complement matrix and it is dense. It corresponds to an elimination of the interior unknowns $\mathring{\mathbf{U}}_i$, $i = 1, 2$.

Multifrontal factorizations

The inverse of A can be easily computed from its factorization

$$A^{-1} = \begin{pmatrix} I & 0 & -A_{11}^{-1}A_{1\Gamma} \\ & I & -A_{22}^{-1}A_{2\Gamma} \\ & & I \end{pmatrix} \begin{pmatrix} A_{11}^{-1} & & \\ & A_{22}^{-1} & \\ & & \mathbb{S}^{-1} \end{pmatrix} \begin{pmatrix} I & & \\ 0 & I & \\ -A_{\Gamma 1}A_{11}^{-1} & -A_{\Gamma 2}A_{22}^{-1} & I \end{pmatrix}. \tag{13}$$

Parallelism:

- Recursion on blocks A_{11} and A_{22} is feasible.
- k-way partitioning

Limitations Schur complement $\mathbb{S}$ is a dense matrix. Factorizing systems of the form

$$\mathbb{S}V_\Gamma = G_\Gamma$$

is a bottleneck.

Schur complement preconditioners

Pros:

- Method of choice if it makes the job
- Genuinely robust black box methods

Cons:

- Worsens beyond a certain size or number of cores (20-30)

From Direct Method to Preconditioners

Instead of factorizing Schur complement, we can solve iteratively systems of the form

$$\mathbb{S} V_\Gamma = G_\Gamma \, .$$

Preconditioner for the Schur complement From a decomposition of matrix $A_{\Gamma\Gamma}$:

$$A_{\Gamma\Gamma} = A_{\Gamma\Gamma}^{(1)} + A_{\Gamma\Gamma}^{(2)}$$

We can infer that for each domain $i = 1, 2$, local operators

$$\mathbb{S}_i := \left(A_{\Gamma\Gamma}^{(i)} - A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma} \right) \quad \text{and } \mathbb{S} = \mathbb{S}_1 + \mathbb{S}_2 \, .$$

and approximate $\mathbb{S}^{-1}$ by

$$\boxed{\mathbb{T} := \frac{1}{2} \left(\mathbb{S}_1^{-1} + \mathbb{S}_2^{-1} \right) \frac{1}{2} \, .}$$

Exact formula if $\mathbb{S}_1 = \mathbb{S}_2 \Longrightarrow$ in general, very good for the high frequency part of the error.

Schur complement preconditioners

Generalizes to many subdomains.
BDD and FETI methods rely on these ideas plus Coarse space corrections.

Pros:

- Very popular in mechanical engineering since it is very efficient for elasticity problems

Cons:

- Needs elementary matrices even for the one-level method
- Not robust for bad decompositions

Outline

Nicolaides Coarse Space

Theorem (Widlund, Dryja)

Let $M_{ASM,2}^{-1}$ be the two-level additive Schwarz method:

$$\kappa(M_{ASM,2}^{-1} A) \leq C \left(1 + \frac{H}{\delta}\right)$$

where δ is the size of the overlap between the subdomains and H the subdomain size.

This does indeed work very well

Number of subdomains	8	16	32	64
ASM	18	35	66	128
ASM + Nicolaides	20	27	28	27

Fails for highly heterogeneous problems
You need a larger and adaptive coarse space.

Failure for Darcy equation with heterogeneities

$$
\begin{aligned}
-\nabla \cdot (\alpha(x,y)\nabla u) &= 0 \quad \text{in} \quad \Omega \subset \mathbb{R}^2, \\
u &= 0 \quad \text{on} \quad \partial\Omega_D, \\
\frac{\partial u}{\partial n} &= 0 \quad \text{on} \quad \partial\Omega_N.
\end{aligned}
$$

Decomposition

$\alpha(x,y)$

Jump	1	10	10^2	10^3	10^4
ASM	39	45	60	72	73
ASM + Nicolaides	30	36	50	61	65

Our approach

Fix the problem by an optimal and proven choice of a coarse space Z.

Fix: Adaptive Coarse Space

Strategy

Define an appropriate coarse space $V_{H2} = \text{span}(Z_2)$ and use the framework previously introduced, writing $R_0 = Z_2^T$ the two level preconditioner is:

$$P_{ASM2}^{-1} := R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i.$$

The coarse space must be

- Local (calculated on each subdomain) $\rightarrow$ parallel
- Adaptive (calculated automatically)
- Easy and cheap to compute
- Robust (must lead to an algorithm whose convergence is proven not to depend on the partition nor the jumps in coefficients)

Adaptive Coarse space for highly heterogeneous Darcy and (compressible) elasticity problems:
Geneo .EVP per subdomain:

Find $V_{j,k} \in \mathbb{R}^{\mathcal{N}_j}$ and $\mu_{j,k} \geq 0$:

$$D_j R_j A R_j^T D_j V_{j,k} = \mu_{j,k} A_j^{Neu} V_{j,k}$$

In the two-level ASM, let τ be a user chosen parameter:
Choose eigenvectors $\mu_{j,k} \geq \tau$ per subdomain:

$$Z := \left(R_j^T D_j V_{j,k} \right)_{\mu_{j,k} \geq \tau}^{j=1,\ldots,N}$$

This automatically includes Nicolaides CS made of Zero Energy Modes.

Adaptive Coarse space for highly heterogeneous Darcy and (compressible) elasticity problems:
Geneo .EVP per subdomain:

Find $V_{j,k} \in \mathbb{R}^{\mathcal{N}_j}$ and $\mu_{j,k} \geq 0$:

$$D_j R_j A R_j^T D_j V_{j,k} = \mu_{j,k} A_j^{Neu} V_{j,k}$$

In the two-level ASM, let τ be a user chosen parameter:
Choose eigenvectors $\mu_{j,k} \geq \tau$ per subdomain:

$$Z := \left(R_j^T D_j V_{j,k} \right)_{\mu_{j,k} \geq \tau}^{j=1,\ldots,N}$$

This automatically includes Nicolaides CS made of Zero Energy Modes.

Theory of GenEO

Two technical assumptions.

Theorem (Spillane, Dolean, Hauret, N., Pechstein, Scheichl (Num. Math. 2013))

If for all j: $0 < \mu_{j,m_{j+1}} < \infty$*:*

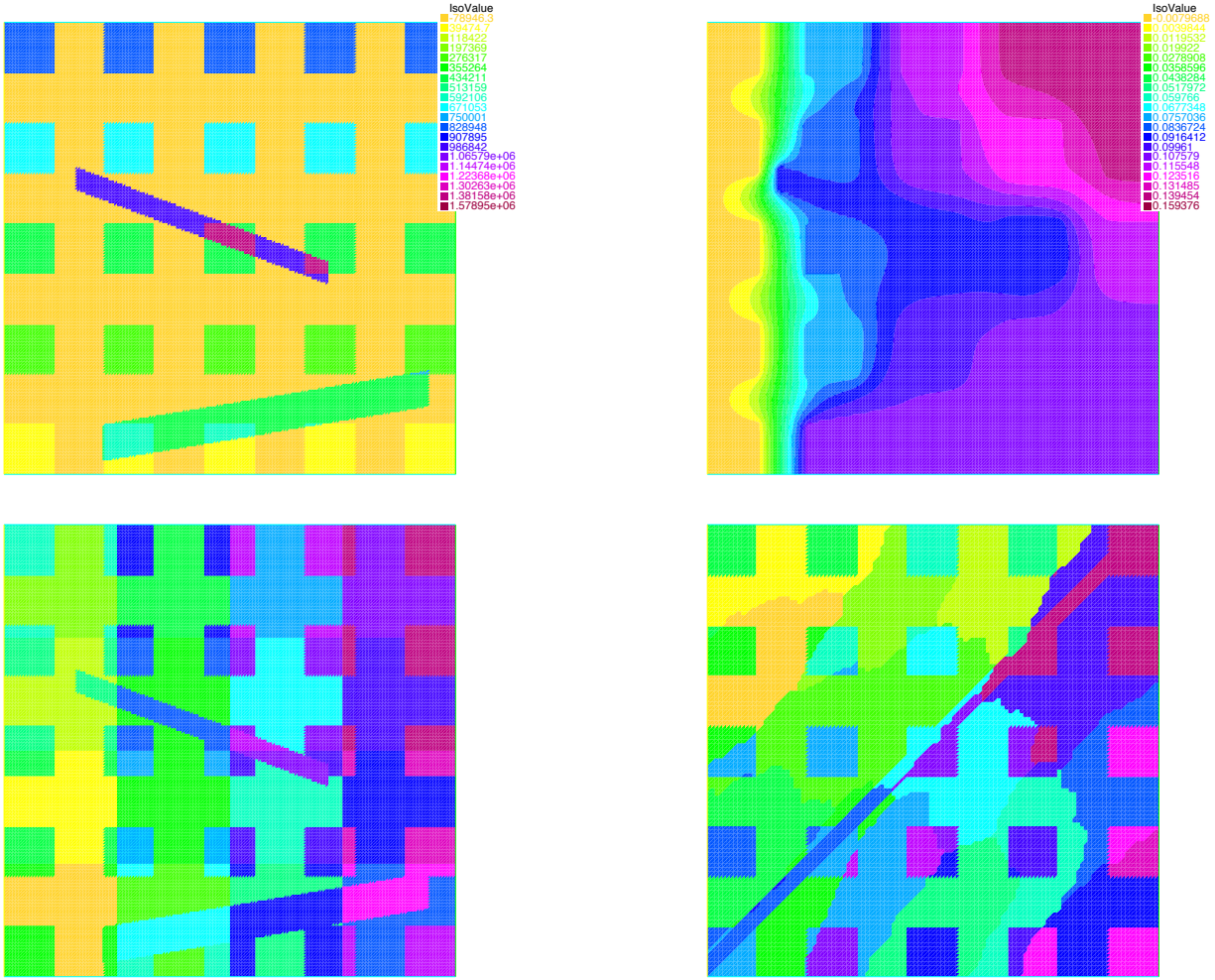
$$\kappa(M_{ASM,2}^{-1}A) \leq (1+k_0)\left[2+k_0(2k_0+1)\left(1+\tau\right)\right]$$

Possible criterion for picking τ: (used in our Numerics)

$$\tau := \min_{j=1,\ldots,N} \frac{H_j}{\delta_j}$$

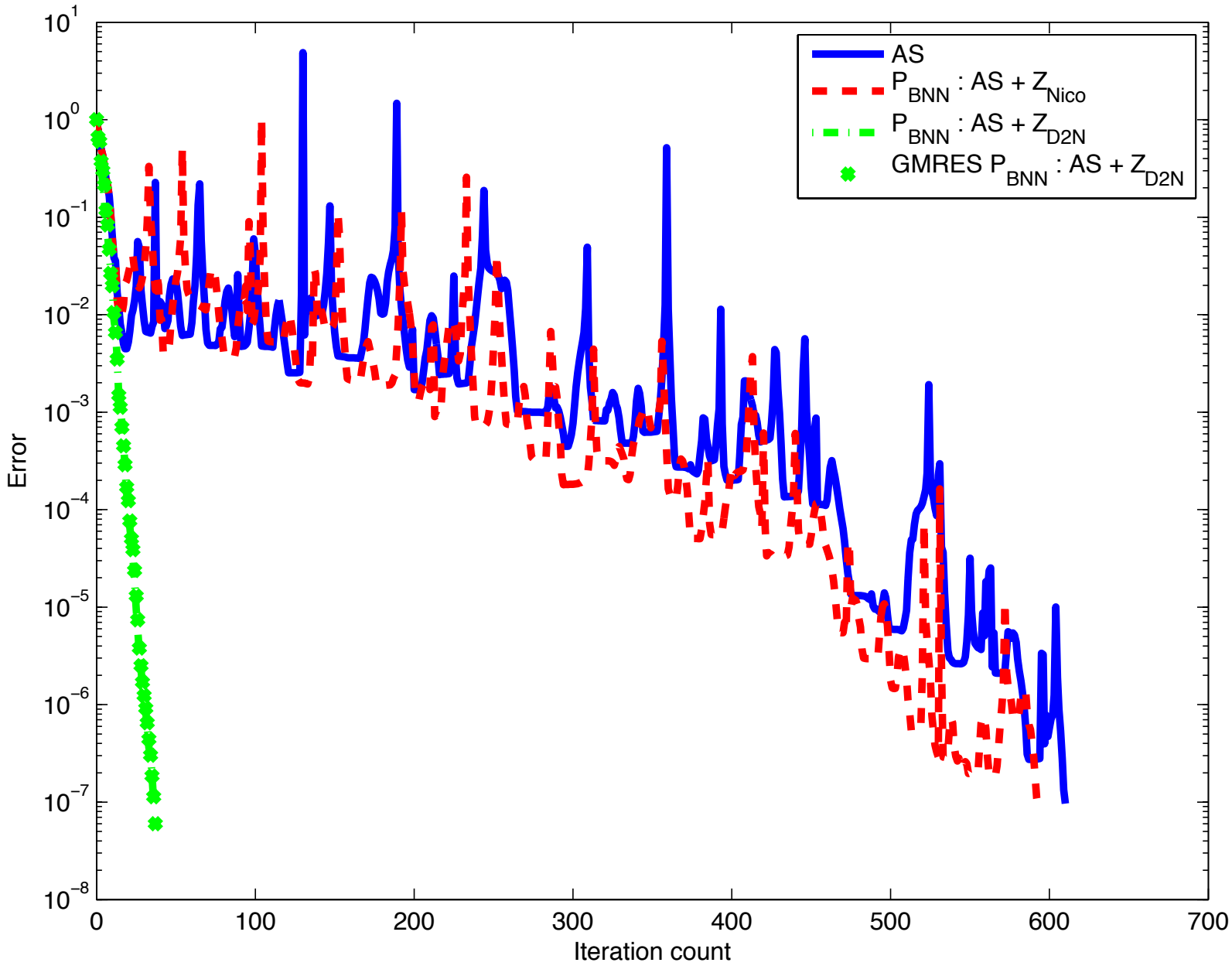
$H_j \ldots$ subdomain diameter, $\delta_j \ldots$ overlap

Numerical results (Darcy)



Channels and inclusions: $1 \leq \alpha \leq 1.5 \times 10^6$, the solution and partitionings (Metis or not)

Convergence



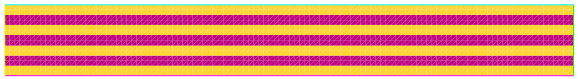
Numerical results – Optimality

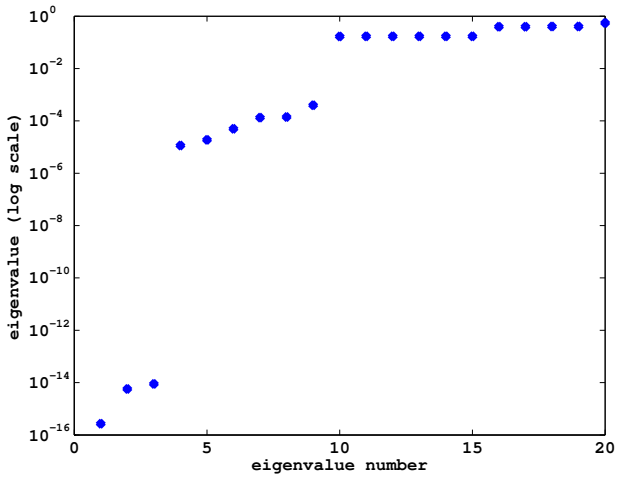
m_i is given automatically by the strategy.

$\#Z$ per subd.	ASM	ASM+Z_{Nico}	ASM+Z_{Geneo}
$\max(m_i - 1, 1)$			273
m_i	614	543	36
$m_i + 1$			32

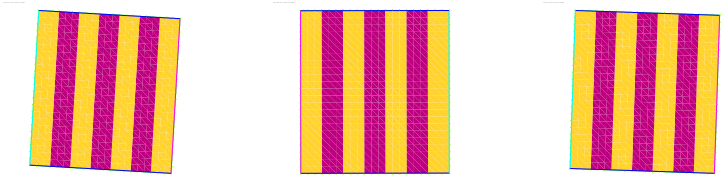
- Taking one fewer eigenvalue has a huge influence on the iteration count
- Taking one more has only a small influence

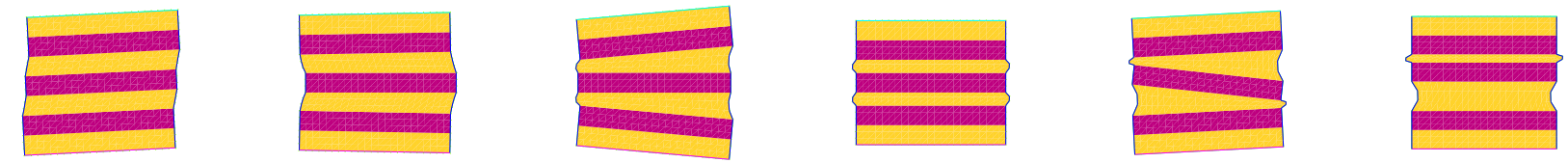
Eigenvalues and eigenvectors (Elasticity)

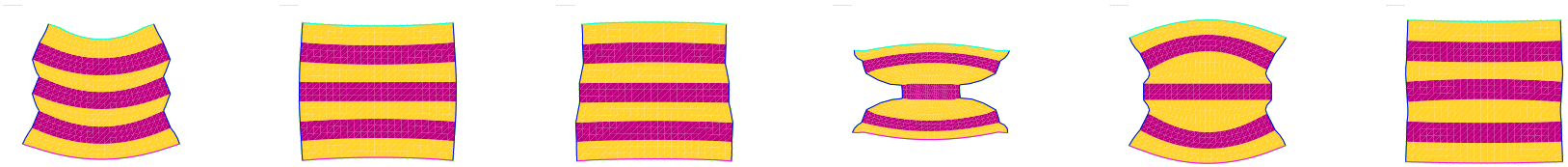




Logarithmic scale







Numerical results via a Domain Specific Language

`FreeFem++` (`http://www.freefem.org/ff++`), with:

- Metis Karypis and Kumar 1998
- SCOTCH Chevalier and Pellegrini 2008
- UMFPACK Davis 2004
- ARPACK Lehoucq et al. 1998
- MPI Snir et al. 1995
- Intel MKL
- PARDISO Schenk et al. 2004
- MUMPS Amestoy et al. 1998
- PaStiX Hénon et al. 2005
- Slepc via PETSC

Runs on PC (Linux, OSX, Windows) and HPC (Babel@CNRS, HPC1@LJLL, Titane@CEA via GENCI PRACE)

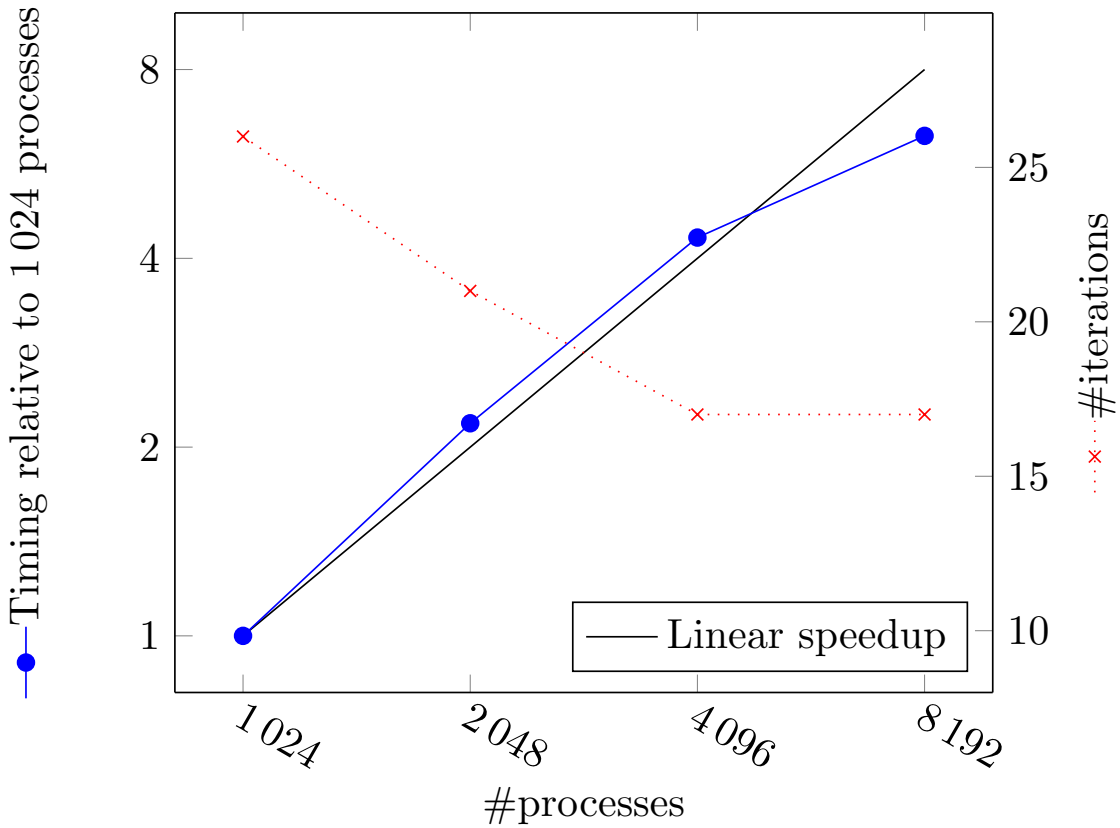
Why use a DS(E)L instead of `C/C++/Fortran/..` ?

- performances close to low-level language implementation,
- hard to beat something as simple as:

$$\begin{aligned}\mathbf{varf}\ \mathrm{a}(\mathrm{u},\ \mathrm{v}) = \mathbf{int3d}(\mathrm{mesh})([\mathbf{dx}(\mathrm{u}),\ \mathbf{dy}(\mathrm{u}),\ \mathbf{dz}(\mathrm{u})]'\ *\ [\mathbf{dx}(\mathrm{v}),\ \mathbf{dy}(\mathrm{v}),\ \mathbf{dz}(\mathrm{v})])\\ +\ \mathbf{int3d}(\mathrm{mesh})(\mathrm{f}\ *\ \mathrm{v})\ +\ \mathbf{on}(\mathrm{boundary\_mesh})(\mathrm{u} = 0)\end{aligned}$$

Strong scalability in two dimensions heterogeneous elasticity (P. Jolivet with Frefeem ++)

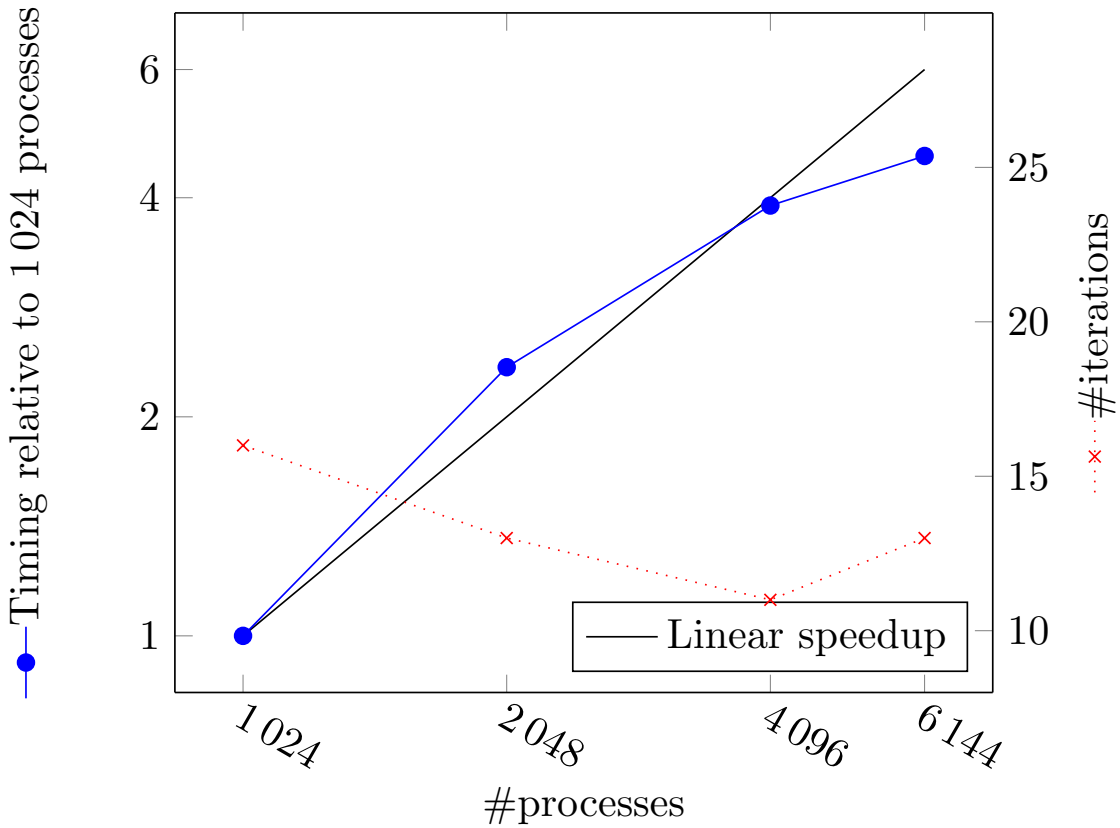
Elasticity problem with heterogeneous coefficients with automatic mesh partition



Speed-up for a 1.2 billion unknowns 2D problem. Direct solvers in the subdomains. Peak performance wall-clock time: 26s.

Strong scalability in three dimensions heterogeneous elasticity

Elasticity problem with heterogeneous coefficients with automatic mesh partition



Speed-up for a 160 million unknowns 3D problem. Direct solvers in subdomains. Peak performance wall-clock time: 36s.

Darcy pressure equation

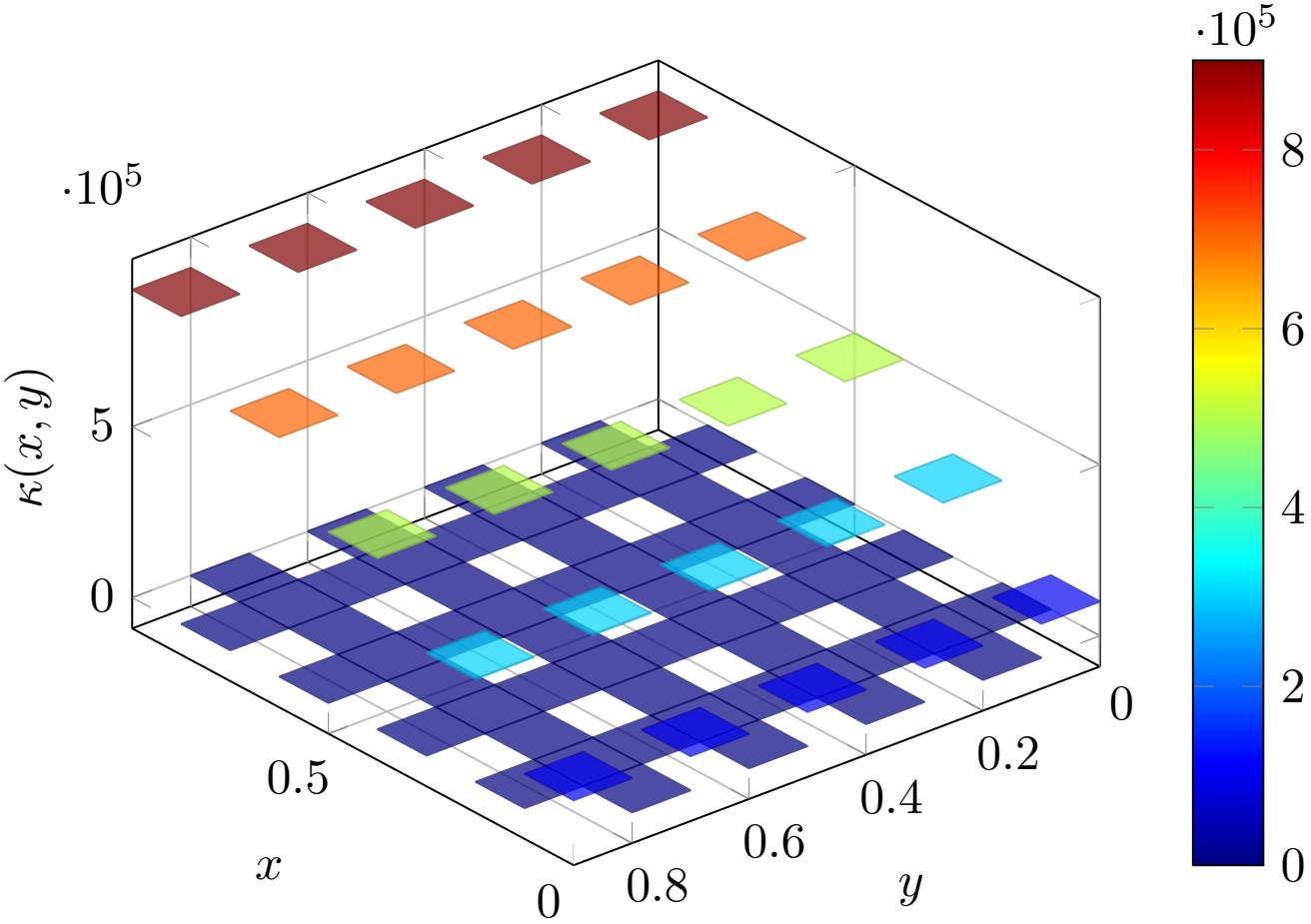
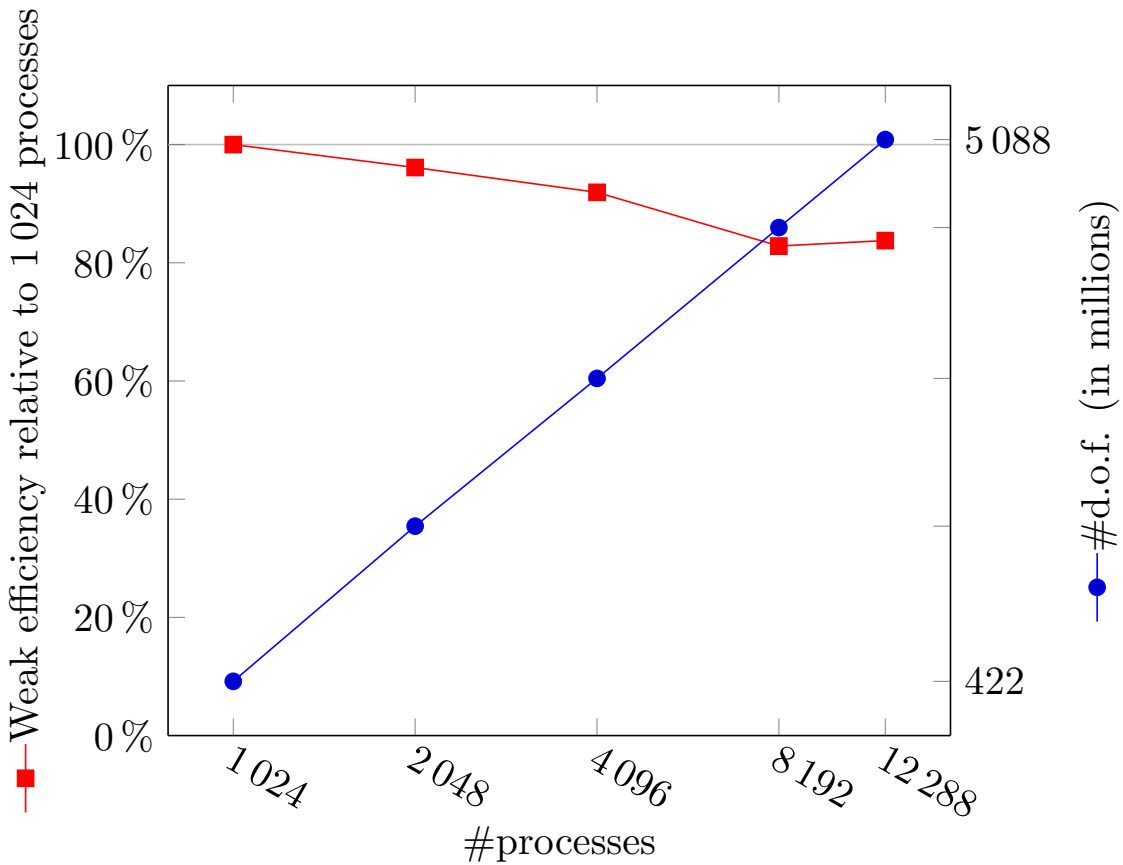


Figure: Two dimensional diffusivity κ

Weak scalability in two dimensions

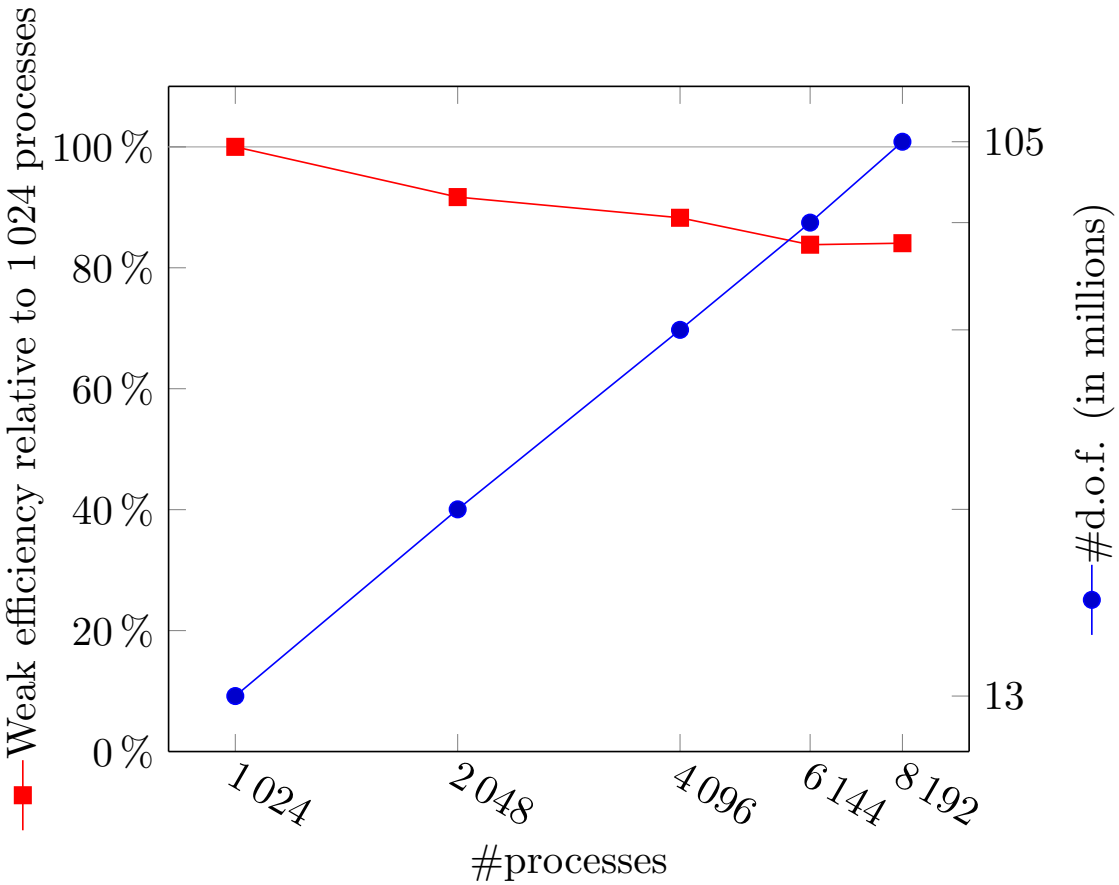
Darcy problems with heterogeneous coefficients with automatic mesh partition



Efficiency for a 2D problem. Direct solvers in the subdomains.
Final size: 22 billion unknowns. Wall-clock time: $\simeq$ 200s.

Weak scalability in three dimensions

Darcy problems with heterogeneous coefficients with automatic mesh partition



Efficiency for a 3D problem. Direct solvers in the subdomains.
Final size: 2 billion unknowns. Wall-clock time: $\simeq$ 200s.

One level ASM revisited

- $H := \mathbb{R}^{\#\mathcal{N}}$ and the a-bilinear form:

$$a(\mathbf{U}, \mathbf{V}) := \mathbf{V}^T A \mathbf{U}. \tag{14}$$

where A is the matrix of the problem we want to solve.

- H_D is a product space and b a bilinear form defined by

$$H_D := \prod_{i=1}^{N} \mathbb{R}^{\#\mathcal{N}_i} \text{ and } b(\mathcal{U}, \mathcal{V}) := \sum_{i=1}^{N} \mathbf{V}_i^T (R_i A R_i^T) \mathbf{U}_i, \ . \tag{15}$$

- The linear operator $\mathcal{R}_{ASM}$ is defined as

$$\mathcal{R}_{ASM} : H_D \longrightarrow H, \ \mathcal{R}_{ASM}(\mathcal{U}) := \sum_{i=1}^{N} R_i^T \mathbf{U}_i. \tag{16}$$

We have: $M_{ASM}^{-1} = \mathcal{R}_{ASM} B^{-1} \mathcal{R}_{ASM}^*$.

Fictitious Space Lemma

Lemma (Fictitious Space Lemma, Nepomnyaschikh 1991)

Let H and H_D be two Hilbert spaces. Let a be a symmetric positive bilinear form on H and b on H_D. Suppose that there exists a linear operator $\mathcal{R} : H_D \to H$, such that

- *$\mathcal{R}$ is surjective.*
- *there exists a positive constant c_R such that*

$$a(\mathcal{R}u_D, \mathcal{R}u_D) \leq c_R \cdot b(u_D, u_D) \quad \forall u_D \in H_D \,. \tag{17}$$

- *Stable decomposition: there exists a positive constant c_T such that for all $u \in H$ there exists $u_D \in H_D$ with $\mathcal{R}u_D = u$ and*

$$c_T \cdot b(u_D, u_D) \leq a(\mathcal{R}u_D, \mathcal{R}u_D) = a(u, u) \,. \tag{18}$$

Fictitious Space Lemma (continued)

Lemma (FSL continued)

We introduce the adjoint operator $\mathcal{R}^* : H \to H_D$ *by* $(\mathcal{R}u_D, u) = (u_D, \mathcal{R}^*u)_D$ *for all* $u_D \in H_D$ *and* $u \in H$. *Then we have the following spectral estimate*

$$c_T \cdot a(u,u) \leq a\left(\mathcal{R}B^{-1}\mathcal{R}^*Au,\, u\right) \leq c_R \cdot a(u,u)\,, \quad \forall u \in H \quad (19)$$

which proves that the eigenvalues of operator $\mathcal{R}B^{-1}\mathcal{R}^*A$ *are bounded from below by* c_T *and from above by* c_R.

This Lemma is the Lax-Milgram lemma of domain decomposition methods.
Combining FSL with GenEO techniques yields an adaptive coarse space with a targeted spectrum for the preconditioned system.

Heuristic motivation for GenEO – I

Let A_i^{Neu} be the Neumann matrix of subdomain i, we define:

$$M_{NN}^{-1} := \sum_{i=1}^{N} R_i^T D_i (A_i^{Neu})^{-1} D_i R_i \,,$$

Lemma

Let k_1 denote the maximum multiplicity of subdomains intersections, then

$$\frac{1}{k_1} \leq \lambda_{min}(M_{NN}^{-1} A) \,.$$

Recall that, (k_0:the maximum number of neighbors)

$$\lambda_{max}(M_{ASM}^{-1} A) \leq k_0 \,.$$

Heuristic motivation for GenEO – II

- One idea would be to blend both preconditioners into a ”perfect” one: No Way
- A second idea is to identify modes $V_{i\,\mu}$ responsible for bad convergence:

$$D_i (A_i^{Neu})^{-1} D_i V_{i\,k} \text{ very different from } A_i^{-1} V_{i\,k}$$

or $\mu_{i\,k}$ far away from 1:

$$A_i^{Neu} V_{i\,k} = \mu_{i\,k} D_i A_i D_i V_{i\,k}$$

Estimate for the one level ASM

Let k_0 be the maximum number of neighbors of a subdomain.
We can take $c_R := k_0$.

Let k_1 be the maximum multiplicity of the intersection between subdomains and τ_1 be defined as:

$$\tau_1 := \min_{1 \leq i \leq N} \min_{U_i \in \mathbb{R}^{\#\mathcal{N}_i} \setminus \{0\}} \frac{\mathbf{U}_i{}^T A_i^{Neu} \mathbf{U}_i}{\mathbf{U}_i{}^T (D_i R_i A R_i^T D_i) \mathbf{U}_i} .$$

We can take $c_T := \frac{\tau_1}{k_1}$.
We have:

$$\frac{\tau_1}{k_1} \leq \lambda(M_{ASM}^{-1} A) \leq k_0 .$$

Control of the lower bound

Definition (Generalized Eigenvalue Problem for the lower bound)

For each subdomain $1 \leq j \leq N$, we introduce the generalized eigenvalue problem

$$\begin{aligned} &\text{Find } (\mathbf{V}_{jk}, \lambda_{jk}) \in \mathbb{R}^{\#\mathcal{N}_j} \setminus \{0\} \times \mathbb{R} \text{ such that} \\ &A_j^{Neu} \mathbf{V}_{jk} = \lambda_{jk} (D_j R_j A R_j^T D_j) \mathbf{V}_{jk} \ . \end{aligned} \tag{20}$$

Let $\tau > 0$ be a user-defined threshold, we define $Z^{\tau}_{geneo,ASM} \subset \mathbb{R}^{\#\mathcal{N}}$ as the vector space spanned by the family of vectors $(R_j^T D_j \mathbf{V}_{jk})_{\lambda_{jk} < \tau, 1 \leq j \leq N}$ corresponding to eigenvalues smaller than τ.

Recap

ASM theory for a S.P.D. matrix A.

(Recap) $A_i := R_i A R_i^T$, $1 \leq i \leq N$

1. Algebraic reformulation $\Rightarrow M_{RAS}^{-1} := \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i$
2. Symmetric variant $\Rightarrow M_{AS}^{-1} := \sum_{i=1}^{N} R_i^T A_i^{-1} R_i$
3. Adaptive Coarse space with prescribed targeted convergence rate
   $\Rightarrow$ Find $V_{j,k} \in \mathbb{R}^{\mathcal{N}_j}$ and $\lambda_{j,k} \geq 0$:

$$D_j R_j A R_j^T D_j V_{j,k} = \lambda_{j,k} A_j^{Neu} V_{j,k}$$

Next develop a similar theory and computational framework for Optimized RAS (ORAS)

Motivation for the Goal

- Fill a ”Hole” in the theoretical framework:
  - No GenEO theory for Adaptive coarse spaces for Optimized interface conditions
  - Whereas it exists for Schwarz and BNN-FETI methods.
- Need for robust methods for nearly incompressible elasticity with arbitrary partitions
  - Combination of ASM with GenEO is very efficient for Darcy and compressible elasticity with arbitrary partitions
  - Combination of BNN-FETI with GenEO is very efficient for Darcy and (in)compressible elasticity with regular partitions

ORAS

Let B_i be the matrix of the Robin subproblem in each subdomain $1 \leq i \leq N$

1. Algebraic reformulation for overlapping subdomains $\Rightarrow$ $M_{ORAS}^{-1} := \sum_{i=1}^{N} R_i^T D_i B_i^{-1} R_i$, *Optimized multiplicative, additive, and restricted additive Schwarz preconditioning, St Cyr et al, 2007*
2. Symmetric variant $\Rightarrow$
   1. $M_{OAS}^{-1} := \sum_{i=1}^{N} R_i^T B_i^{-1} R_i$ (Natural but K.O.)
   2. $M_{SORAS}^{-1} := \sum_{i=1}^{N} R_i^T D_i B_i^{-1} D_i R_i$ (O.K.)
3. Adaptive Coarse space with prescribed targeted convergence rate
   $\Rightarrow$??

FSL and one level SORAS

- $H := \mathbb{R}^{\#\mathcal{N}}$ and the a-bilinear form:

$$a(\mathbf{U}, \mathbf{V}) := \mathbf{V}^T A \mathbf{U}. \tag{21}$$

  where A is the matrix of the problem we want to solve.
- H_D is a product space and b a bilinear form defined by

$$H_D := \prod_{i=1}^{N} \mathbb{R}^{\#\mathcal{N}_i} \text{ and } b(\mathcal{U}, \mathcal{V}) := \sum_{i=1}^{N} \mathbf{V}_i^T B_i \mathbf{U}_i, \ . \tag{22}$$

- The linear operator $\mathcal{R}_{SORAS}$ is defined as

$$\mathcal{R}_{SORAS} : H_D \longrightarrow H, \ \mathcal{R}_{SORAS}(\mathcal{U}) := \sum_{i=1}^{N} R_i^T D_i \mathbf{U}_i. \tag{23}$$

We have: $M_{SORAS}^{-1} = \mathcal{R}_{SORAS} B^{-1} \mathcal{R}_{SORAS}^*$.

Estimate for the one level SORAS

Let k_0 be the maximum number of neighbors of a subdomain and γ_1 be defined as:

$$\gamma_1 := \max_{1 \leq i \leq N} \max_{\mathbf{U}_i \in \mathbb{R}^{\#\mathcal{N}_i} \setminus \{0\}} \frac{\left(R_i^T D_i \mathbf{U}_i\right)^T A \left(R_i^T D_i \mathbf{U}_i\right)}{\mathbf{U}_i^T B_i \mathbf{U}_i}$$

We can take $c_R := k_0 \, \gamma_1$.
Let k_1 be the maximum multiplicity of the intersection between subdomains and τ_1 be defined as:

$$\tau_1 := \min_{1 \leq i \leq N} \min_{U_i \in \mathbb{R}^{\#\mathcal{N}_i} \setminus \{0\}} \frac{{\mathbf{U}_i}^T A_i^{Neu} \mathbf{U}_i}{{\mathbf{U}_i}^T B_i \mathbf{U}_i} \, .$$

We can take $c_T := \frac{\tau_1}{k_1}$.
We have:

$$\frac{\tau_1}{k_1} \leq \lambda(M_{SORAS}^{-1} A) \leq k_0 \, \gamma_1 \, .$$

Control of the upper bound

Definition (Generalized Eigenvalue Problem for the upper bound)

$$\text{Find } (\mathbf{U}_{ik}, \mu_{ik}) \in \mathbb{R}^{\#\mathcal{N}_i} \setminus \{0\} \times \mathbb{R} \text{ such that}$$
$$D_i R_i A R_i^T D_i \mathbf{U}_{ik} = \mu_{ik} B_i \mathbf{U}_{ik} \quad . \tag{24}$$

Let $\gamma > 0$ be a user-defined threshold, we define $Z^{\gamma}_{geneo} \subset \mathbb{R}^{\#\mathcal{N}}$ as the vector space spanned by the family of vectors $(R_i^T D_i \mathbf{U}_{ik})_{\mu_{ik} > \gamma, 1 \leq i \leq N}$ corresponding to eigenvalues larger than γ.

Control of the lower bound

Definition (Generalized Eigenvalue Problem for the lower bound)

For each subdomain $1 \leq j \leq N$, we introduce the generalized eigenvalue problem

$$\begin{aligned} &\text{Find } (\mathbf{V}_{jk}, \lambda_{jk}) \in \mathbb{R}^{\#\mathcal{N}_j} \setminus \{0\} \times \mathbb{R} \text{ such that} \\ &A_j^{Neu} \mathbf{V}_{jk} = \lambda_{jk} B_j \mathbf{V}_{jk} \ . \end{aligned} \tag{25}$$

Let $\tau > 0$ be a user-defined threshold, we define $Z_{geneo}^{\tau} \subset \mathbb{R}^{\#\mathcal{N}}$ as the vector space spanned by the family of vectors $(R_j^T D_j \mathbf{V}_{jk})_{\lambda_{jk} < \tau, 1 \leq j \leq N}$ corresponding to eigenvalues smaller than τ.

Two level SORAS-GENEO-2 preconditioner

Definition (Two level SORAS-GENEO-2 preconditioner)

Let P_0 denote the a-orthogonal projection on the SORAS-GENEO-2 coarse space

$$Z_{\text{GenEO-2}} := Z_{geneo}^{\tau} \bigoplus Z_{geneo}^{\gamma} ,$$

the two-level SORAS-GENEO-2 preconditioner is defined:

$$\boxed{M_{SORAS,2}^{-1} := P_0 A^{-1} + (I_d - P_0)\, M_{SORAS}^{-1} (I_d - P_0^T)}$$

where $P_0 A^{-1} = R_0^T (R_0\, A\, R_0^T)^{-1} R_0$, see J. Mandel, 1992.

Two level SORAS-GENEO-2 preconditioner

Theorem (Haferssas, Jolivet and N., 2015)

Let γ and τ be user-defined targets. Then, the eigenvalues of the two-level SORAS-GenEO-2 preconditioned system satisfy the following estimate

$$\frac{1}{1+\frac{k_1}{\tau}} \leq \lambda(M_{SORAS,2}^{-1} A) \leq \max(1, k_0\,\gamma)$$

What if one level method is M_{OAS}^{-1}:

$$\text{Find } (\mathbf{V}_{jk}, \lambda_{jk}) \in \mathbb{R}^{\#\mathcal{N}_i} \setminus \{0\} \times \mathbb{R} \text{ such that}$$
$$A_i^{Neu} \mathbf{V}_{ik} = \lambda_{ik} D_i B_i D_i \mathbf{V}_{ik} \ .$$

Nearly incompressible elasticity

Material properties: Young modulus E and Poisson ratio ν or alternatively by its Lamé coefficients λ and μ:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{and} \quad \mu = \frac{E}{2(1+\nu)}.$$

For ν close to $1/2$, the variational problem consists in finding $(\boldsymbol{u}_h, p_h) \in \mathcal{V}_h := \mathbb{P}_2^d \cap H_0^1(\Omega) \times \mathbb{P}_1$ such that for all $(\boldsymbol{v}_h, q_h) \in \mathcal{V}_h$

$$\begin{cases} \int_\Omega 2\mu \underline{\underline{\varepsilon}}(\boldsymbol{u}_h) : \underline{\underline{\varepsilon}}(\boldsymbol{v}_h) dx & - \int_\Omega p_h \mathrm{div}\,(\boldsymbol{v}_h) dx = \int_\Omega \boldsymbol{f}\boldsymbol{v}_h dx \\ - \int_\Omega \mathrm{div}\,(\boldsymbol{u}_h) q_h dx & - \int_\Omega \frac{1}{\lambda} p_h q_h = 0 \end{cases}$$

$$\Longrightarrow A\mathbf{U} = \begin{bmatrix} H & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathrm{f} \\ 0 \end{bmatrix} = \mathbf{F}.$$

A is symmetric but no longer positive.

Comparisons (with FreeFem++)

Figure: 2D Elasticity: Sandwich of steel $(E_1, \nu_1) = (210 \cdot 10^9, 0.3)$ and rubber $(E_2, \nu_2) = (0.1 \cdot 10^9, 0.4999)$.

Metis partitioning

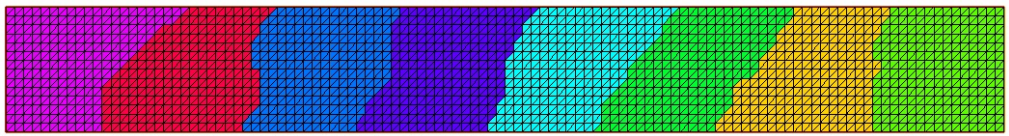


Table: 2D Elasticity. GMRES iteration counts

		AS	SORAS	AS+CS(ZEM)		SORAS +CS(ZEM)		AS-GenEO		SORAS -GenEO-2	
Nb DOFs	Nb subdom	iteration	iteration	iteration	*dim*	iteration	*dim*	iteration	*dim*	iteration	*dim*
35841	8	150	184	117	*24*	79	*24*	110	*184*	13	*145*
70590	16	276	337	170	*48*	144	*48*	153	*400*	17	*303*
141375	32	497	++1000	261	*96*	200	*96*	171	*800*	22	*561*
279561	64	++1000	++1000	333	*192*	335	*192*	496	*1600*	24	*855*
561531	128	++1000	++1000	329	*384*	400	*384*	++1000	*2304*	29	*1220*
1077141	256	++1000	++1000	369	*768*	++1000	*768*	++1000	*3840*	36	*1971*

Strong scalability in two and three dimensions (with FreeFem++ and HPDDM)

Stokes problem with automatic mesh partition. Driven cavity problem

	N	Factorization	Deflation	Solution	# of it.	Total	# of d.o.f.
3D	1 024	79.2 s	229.0 s	76.3 s	45	387.5 s	$50.63 \cdot 10^6$
	2 048	29.5 s	76.5 s	34.8 s	42	143.9 s	
	4 096	11.1 s	45.8 s	19.8 s	42	80.9 s	
	8 192	4.7 s	26.1 s	14.9 s	41	56.8 s	
2D	1 024	5.2 s	37.9 s	51.5 s	51	95.6 s	$100.13 \cdot 10^6$
	2 048	2.4 s	19.3 s	22.1 s	42	44.5 s	
	4 096	1.1 s	10.4 s	10.2 s	35	22.6 s	
	8 192	0.5 s	4.6 s	6.9 s	38	12.7 s	

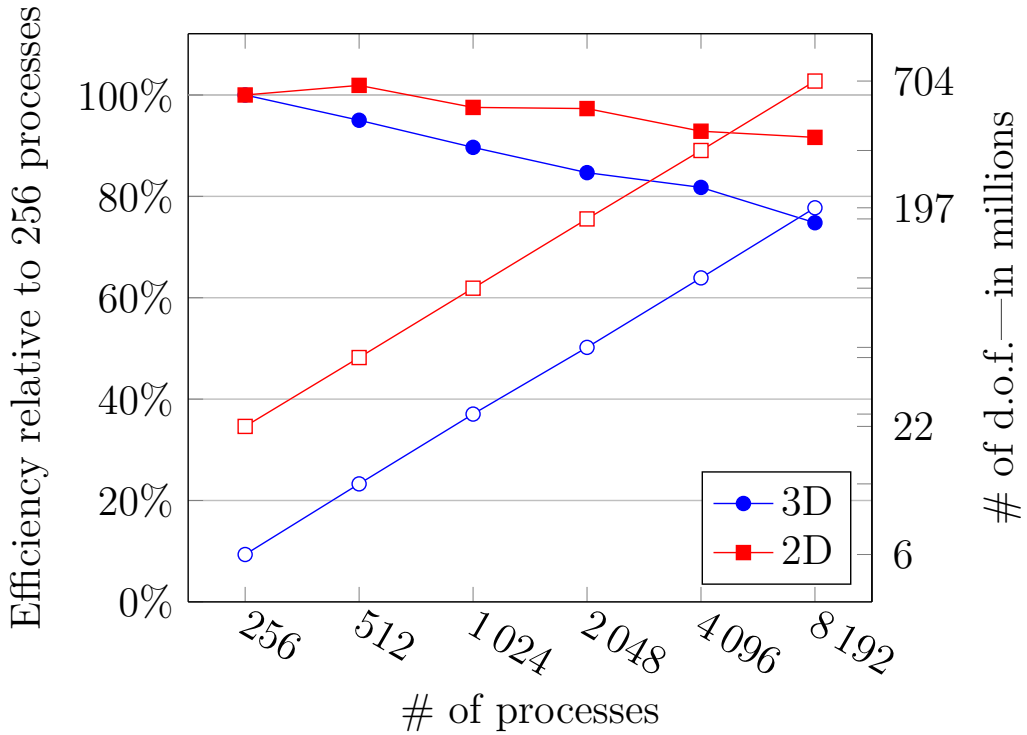
Peak performance: 50 millions d.o.f's in 3D in 57 sec.

IBM/Blue Gene Q machine with 1.6 GHz Power A2 processors.

Hours provided by an IDRIS-GENCI project.

Weak scalability for heterogeneous elasticity (with FreeFem++ and HPDDM)

Rubber Steel sandwich with automatic mesh partition



(a) Timings of various simulations

200 millions unknowns in 3D wall-clock time: 200. sec.
IBM/Blue Gene Q machine with 1.6 GHz Power A2 processors.
Hours provided by an IDRIS-GENCI project.

Outline

HPDDM Library (P. Jolivet and N.)

An implementation of several Domain Decomposition Methods

- One-and two-level Schwarz methods
- The Finite Element Tearing and Interconnecting (FETI) method
- Balancing Domain Decomposition (BDD) method

Library

- Linked with BLAS & LAPACK.
- Linked with state of the art solvers: direct solvers (MUMPS, SuiteSparse, MKL PARDISO, PASTIX), multigrid: BoomerAMG
- Linked with eigenvalue solver (ARPACK).
- Interfaced with discretization kernel FreeFem++ & FEEL++
- C++, C, Python and Fortran interface

HPDDM Examples

FreeFem++

- `examples++-hpddm` diffusion, elasticity, heat, Helmholtz, Maxwell
- Schwarz or FETI-BDD (diffusion-elasticity only)

Feel++ (C. Prud'homme)

- `doc/manual/dd/geneo.cpp` FETI-BDD + Geneo
- `doc/manual/ns/nsprojRecycling.cpp` Navier-Stokes with projection + GCRODR

Stand alone `examples`

- Schwarz with Geneo finite difference (C++,C Python)
- Krylov methods in Fortran (Block and Recycling)
- Schwarz from file (CSR format)
- Soon : Krylov methods with Petsc preconditioners (Block and Recycling)

some FreeFem++ code: Schwarz.edp

```
mesh Th = minimalMesh;
func Pk = P1;
fespace Wh(Th, Pk);
// Mesh decomposition and distribution among processes
build(generateTh, Th, ThBorder, ThOverlap, s, D,
    numberIntersection, arrayIntersection,
    restrictionIntersection, Wh, Pk, mpiCommWorld)
// Variational formulation of the problem
macro Varf(varfName, meshName, PhName)
    varf varfName(u, v) = intN(meshName)((grad(u)' * grad(v))) +
        intN(meshName)(v) + on(1, u = 0.0);// EOM
// Distributed matrix A
assemble(Mat, rhs, Wh, Th, ThBorder, Varf)
dschwarz A(Mat, arrayIntersection, restrictionIntersection,
    scaling = D);
```

some FreeFem++ code: Schwarz.edp

```
// Geneo coarse space construction
macro EVproblem(varfName, meshName, PhName)
varf varfName(u, v) = intN(meshName)((grad(u)' * grad(v))) + on
    (1, u = 0.0);// EOM
EVproblem(vPbNoPen, Th, Ph)
matrix<real> noPen = vPbNoPen(Wh, Wh, solver = CG);
attachCoarseOperator(mpiCommWorld, A, A = noPen);
// DDM solve
Wh<real> def(u); // this will be the solution
DDM(A, u[], rhs);
plotMPI(Th, u[], "Global solution", Pk, def, 3, 1)
```

Outline

Convection-Diffusion Equation

More General Interface Conditions for the Convection-Diffusion Equation

$$cu + \vec{a} \cdot \nabla u - \nu \Delta u = f$$

where $c = 1/\Delta t$ if a backward Euler scheme is used.

The analysis of the convergence reveals that interface conditions with second order derivatives must have the following form

$$\partial_n + \frac{\vec{a} \cdot n}{2\nu} + \alpha + \beta \partial_\tau - \gamma \partial^2_{\tau^2}$$

with $\alpha, \gamma > 0$ and where n is the outward normal to the subdomain and τ is tangent to the interface.

Convection-Diffusion Equation

For a constant coefficient operator, the convergence rate in the Fourier space is given by

$$\rho(k;\alpha,\beta,\gamma) \equiv \left| \frac{\frac{\sqrt{(\vec{a}\cdot n)^2 + 4\nu c + 4I\vec{a}\cdot\tau\nu k + 4\nu^2k^2}}{2\nu} - (\alpha + I\beta k + \gamma k^2)}{\frac{\sqrt{(\vec{a}\cdot n)^2 + 4\nu c + 4I\vec{a}\cdot\tau\nu k + 4\nu^2k^2}}{2\nu} + (\alpha + I\beta k + \gamma k^2)} \right|$$

where $I^2 = -1$.

Convection-Diffusion Equation

Possible choices for α, β and γ:

- *Exact* absorbing boundary conditions: limited to constant coefficient operators (T. Hagström et al., 1988)
- $\beta = \gamma = 0$ (Quarteroni, 1996).
- *Approximate* absorbing boundary conditions of order 0, 1 or 2, referred to as Taylor of order 0,1 or 2, (N. & F. Rogier, 1992).
- α as in (N. & F. Rogier, 1992), *optimization* over β and γ (C. Japhet, N. & F. Rogier).

Schur Complement type method for Convection-Diffusion problem

Robin-Robin Algorithm
Y.A, P. Letallec, F. Nataf, M. Vidrascu
Schur method for a two subdomain case
Find an equation whose solution is the interface value of u on interface Γ.

Consider first Dirichlet local sub problems with $u|_\Gamma$ on the interface.

$$\begin{aligned} \mathcal{L}u_i &= f \quad \text{in } \Omega_i, \\ u_i &= u|_\Gamma \quad \text{on } \Gamma \\ u_i &= 0 \quad \text{sur } \partial\Omega_i \backslash \Gamma. \end{aligned}$$

Consider the DtN (Dirichlet to Neumann a.k.a. Steklov-Poincaré) map:

$$\begin{aligned} \mathcal{S} &: L^2(\Omega) \times H_{00}^{\frac{1}{2}}(\Gamma) \to H^{-\frac{1}{2}}(\Gamma) \\ \mathcal{S}(f, u|_\Gamma) &= \frac{\nu}{2}\left(\frac{\partial u_1}{\partial n_1} + \frac{\partial u_2}{\partial n_2}\right)|_\Gamma \end{aligned}$$

The interface problem reads

$$\mathcal{S}(0, u|_\Gamma) = -\mathcal{S}(f, 0)$$

After discretization, the problem is solved by a preconditioned GMRES solver.

Neumann-Neumann Algorithm

(R Glowinski, P. Letallec et al)
The basic idea to approximte the inverse of $\mathcal{S}(0,.)$ by $\mathcal{T}$:

$$\mathcal{T} : H^{-\frac{1}{2}}(\Gamma) \rightarrow H^{\frac{1}{2}}_{00}(\Gamma)$$
$$\mathcal{T}(g) = \frac{1}{2}\left(v_1 + v_2\right)|_\Gamma.$$

where

$$\begin{aligned} &\mathcal{L}v_i = 0 \quad \text{in } \Omega_i, \\ &-\nu\frac{\partial v_i}{\partial n_i} = g \quad \text{on } \Gamma \\ &v_i = 0 \quad \text{sur } \partial\Omega_i \backslash \Gamma. \end{aligned}$$

Remark In the symmetric two subdomain case, and if $\vec{a}$ is uniform and $\vec{a}\cdot\vec{n} = 0$, then $\mathcal{T}$ **is the exact inverse of** $\mathcal{S}(0,.)$**.**
Remark General case: if $\vec{a}\cdot\vec{n} = 0$, **this preconditioner is nearly optimal if a coarse space is added (multigrid ingredient).**
Remark when $\vec{a}\cdot\vec{n} \neq 0$, Neumann-Neumann is too much symmtric.

Robin-Robin Algorithm

The basic idea is to define the preconditioner $\mathcal{T}$ as:

$$\mathcal{T} : H^{-\frac{1}{2}}(\Gamma) \rightarrow H_{00}^{\frac{1}{2}}(\Gamma)$$
$$\mathcal{T}(g) = \frac{1}{2}\left(v_1 + v_2\right)|_\Gamma.$$

where

$$\begin{aligned} &\mathcal{L} v_i = 0 \quad \text{in } \Omega_i, \\ &\nu \frac{\partial v_i}{\partial n_i} - \frac{\vec{a}\cdot\vec{n}_i}{2} v_i = g \quad \text{on } \Gamma \\ &v_i = 0 \quad \text{on } \partial\Omega_i \backslash \Gamma. \end{aligned}$$

Remark Sub problems are well posed.
Remark Generalizes Neumann-Neumann algorithm when $a \cdot n \neq 0$.

Important remark

The convection diffusion equation reads

$$(c - \frac{1}{2}\nabla \cdot \vec{a})u + \frac{1}{2}(\vec{a} \cdot \nabla u + \nabla \cdot (\vec{a}u)) - \nu \Delta u = f$$

It can be written in a skew-symmetric form:

$$\int_{\Omega_i} \left\{ \nu \nabla u \cdot \nabla v + \frac{1}{2}(\vec{a} \cdot \nabla u)\, v - \frac{1}{2}(\vec{a} \cdot \nabla v)\, u + (c - \frac{1}{2}\nabla \cdot (\vec{a}))u\, v \right\}$$

The above Robin condition is the natural boundary condition of this variational formulation.
Similarity with Neumann-Neumann algorithm for the Laplacian.
Consequence Starting from a Neumann-Neumann code, the modification is easy. **For SUPG stabilization, the preconditioner is automatically setup.**
Remark Skew symmetric form plays a role only for interface nodes.

Convergence analysis for simple cases

Two half planes case
proof : Fourier

$$\widehat{\mathcal{S}}(\xi) = \widehat{\mathcal{T}}^{-1}(\xi) = \frac{1}{2}\sqrt{4\nu c + (\vec{a}.\vec{n}_k)^2 + 4i\vec{a}.\vec{\tau}_2\xi\nu + 4\xi^2\nu^2}.$$

For more slices, if

$$\max(\frac{cL}{|\vec{a}.\vec{n}|}, L\sqrt{\frac{c}{\nu}}) \gg 1,$$

the,

$$\mathcal{T} \circ \mathcal{S} \simeq Id.$$

Good behvior for an advective term not too large or for a small viscosity.

Plus de deux sous-domaines avec convection forte

Cas typique

- grand pas de temps $\Rightarrow c \ll 1$ ou $c = 0$.
- ν arbitraire.
- on suppose la vitesse uniforme.
- Les sous-domaines sont des bandes et on suppose $\vec{a} \cdot \vec{n} \neq 0$ sur les interfaces. Le nombre de bandes est N.

Premiers tests: comparaisons - I

Rectangle $[0,1] \times [0,0.2]$ découpé en cinq bandes 0.2×0.2.
Dans chaque sous-domaine, maillage uniforme 60×60 éléments.
paramètres : $c = 1$ et $\nu = 0.001$ ou $\nu = 1$, et quatre vitesses :

1. $\vec{a} = \vec{e_1}$.
   la vitesse est ici normale aux interfaces
2. $\vec{a} = \vec{e_2}$.
   la vitesse est ici parallèle aux interfaces
3. $\vec{a} = \frac{\sqrt{2}}{2}(\vec{e_1} + \vec{e_2})$.
   vitesse oblique.
4. $\vec{a} = 2\pi \left((x_1 - 0.5)\vec{e_2} - (x_2 - 0.1)\vec{e_1} \right)$.
   vitesse tournante : tourbillon.

Premiers tests: comparaisons - II

Table: comparaisons pour différents champs de vitesse

viscosité	Precond.\ vitesse.	$\perp$	//	oblique	tournante
$\nu = 0.001$	R-R	3	2	5	36
	N-N	52	2	42	> 100
	–	14	34	13	71
$\nu = 1$	R-R	9	9	10	10
	N-N	9	9	10	11
	–	30	38	41	41

- La méthode de **Robin-Robin est bien meilleure** que celle de Neumann-Neumann quand **la viscosité est petite** alors que les **performances sont équivalentes pour** ν **grand.**
- pour $\nu \ll 1$, et si la vitesse n'est pas $//$ aux interfaces, la méthode de Neumann-Neumann est très mauvaise.
  Au contraire, si la vitesse est $//$, alors Neumann-Neumann et Robin-Robin sont équivalents et sont quasi optimaux (2 iterations).
- Accord complet avec l'analyse de Fourier.
- Donc la méthode de Robin-Robin **s'adapte automatiquement** aux différents cas.
- la méthode de Robin-Robin est toujours la moins chère.

Open questions for the non symmetric case

- Adaptive interface construction
- Adaptive coarse space constructions
- Non symmetric extension of the Fictitious Space Lemma

Outline

Conclusion

Summary

- Using two generalized eigenvalue problems and projection preconditioning we are able to achieve a targeted convergence rate for
  - Additive Schwarz method (ASM)
  - Optimized Schwarz method
  - BNN methods (see Lecture Notes)
- Available in HPDDM C++/MPI library
- Available in the public release of FreeFem++

Conclusion

Future work

- Build the coarse space on the fly, see e.g. M. Szydlarski (2013), N. Spillane (2016).
- Nonlinear time dependent problem (Reuse of the coarse space)
- Multigrid like three (or more) level methods
- Coarse spaces for non symmetric, undefinite problems

Bibliography Coarse Space for Schwarz

Preprints available on HAL and Software on freefem.org and github:


P. Jolivet, V. Dolean, F. Hecht, F. Nataf, C. Prud'homme, N. Spillane, "High Performance domain decomposition methods on massively parallel architectures with FreeFem++", J. of Numerical Mathematics, 2012 vol. 20.


N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, R. Scheichl, "Abstract Robust Coarse Spaces for Systems of PDEs via Generalized Eigenproblems in the Overlaps", Numerische Mathematik, 2013.


R. Haferssas, P. Jolivet and F Nataf, "A robust coarse space for Optimized Schwarz methods SORAS-GenEO-2", https://hal.archives-ouvertes.fr/hal-01100926 , 2015, submitted.


V. Dolean, P. Jolivet and F Nataf, "An Introduction to Domain Decomposition Methods: algorithms, theory and parallel implementation", SIAM, 2015.


P. Jolivet and F Nataf, "HPDDM: high-performance unified framework for domain decomposition methods", https://github.com/hpddm/hpddm , MPI-C++ library, 2014.

THANK YOU FOR YOUR ATTENTION!

Bibliography Coarse Space for Schwarz

Preprints available on HAL and Software on freefem.org and github:


P. Jolivet, V. Dolean, F. Hecht, F. Nataf, C. Prud'homme, N. Spillane, "High Performance domain decomposition methods on massively parallel architectures with FreeFem++", J. of Numerical Mathematics, 2012 vol. 20.


N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, R. Scheichl, "Abstract Robust Coarse Spaces for Systems of PDEs via Generalized Eigenproblems in the Overlaps", Numerische Mathematik, 2013.


R. Haferssas, P. Jolivet and F Nataf, "A robust coarse space for Optimized Schwarz methods SORAS-GenEO-2", https://hal.archives-ouvertes.fr/hal-01100926 , 2015, submitted.


V. Dolean, P. Jolivet and F Nataf, "An Introduction to Domain Decomposition Methods: algorithms, theory and parallel implementation", SIAM, 2015.


P. Jolivet and F Nataf, "HPDDM: high-performance unified framework for domain decomposition methods", https://github.com/hpddm/hpddm , MPI-C++ library, 2014.

THANK YOU FOR YOUR ATTENTION!