GENEO: une méthode de décomposition de domaine à deux niveaux pour des systèmes d’équations très hétérogènes

Frédéric Nataf

Laboratory J.L. Lions, Univ. Paris VI and CNRS

joint work with
Nicole Spillane (Univ. Paris VI)
Victorita Dolean (Univ. Nice Sophia-Antipolis)
Patrice Hauret (Michelin, Clermont-Ferrand)
Clemens Pechstein (Johannes Kepler Univ., Linz)
Robert Scheichl (Univ. Bath)

GENEO – ONERA/DSNA 2012
Outline

1. Introduction
2. Coarse space for heterogeneous problems: the DtN algorithm
3. An abstract 2-level Schwarz: the GenEO algorithm
4. Bibliography
**Motivation**

Large discretized system of PDEs
strongly heterogeneous coefficients
(high contrast, nonlinear, multiscale)

E.g. Darcy pressure equation,
\( P^1 \)-finite elements:

\[
AU = F
\]

\[
\text{cond}(A) \sim \frac{\alpha_{\text{max}}}{\alpha_{\text{min}}} h^{-2}
\]

**Goal:**
iterative solvers
robust in size and heterogeneities

**Applications:**
flow in heterogeneous / stochastic / layered media
structural mechanics
electromagnetics etc.
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.
### Linear Algebra from the End User point of view

#### Direct
- **Cons:** Memory
- **Difficult to solve (MAT, RHS, SOL)**
- **Pros:** Robustness

#### DDM
- **Pro:** Flexible
- **Naturally solve (MAT, RHS, SOL)**
- **Few black box routines**
- **Few implementations of efficient DDM**

#### Iterative
- **Pros:** Memory
- **Easy to solve (MAT, RHS, SOL)**
- **Cons:** Robustness

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

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

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

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

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

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

Parallel algorithm, converges but very slowly, overlapping subdomains only.
The parallel version is called Jacobi Schwarz method (JSM).
The algorithm acts on the local functions \( (u_i)_{i=1,2} \).

To make things global, we need:

- **extension operators**, \( E_i \), s.t. for a function \( w_i : \Omega_i \mapsto \mathbb{R} \), \( E_i(w_i) : \Omega \mapsto \mathbb{R} \) is the extension of \( w_i \) by zero outside \( \Omega_i \).

- **partition of unity functions** \( \chi_i : \Omega_i \mapsto \mathbb{R} \), \( \chi_i \geq 0 \) and \( \chi_i(x) = 0 \) for \( x \in \partial \Omega_i \) and s.t.

\[
\sum_{i=1}^{2} E_i(\chi_i \, w_{|\Omega_i}) = w
\]

Let \( u^n \) be an approximation to the solution to the global Poisson problem and \( u^{n+1} \) is computed by solving first local subproblems and then gluing them together.
Local problems to solve

\[- \Delta (u_{i}^{n+1}) = f \quad \text{in} \quad \Omega_{i} \]
\[u_{i}^{n+1} = 0 \quad \text{on} \quad \partial\Omega_{i} \cap \partial\Omega \]
\[u_{i}^{n+1} = u^{n} \quad \text{on} \quad \partial\Omega_{i} \cap \overline{\Omega_{3-i}}.\]

Two ways to "glue" solutions

● Using the partition of unity functions
  
  **Restricted Additive Schwarz (RAS)**

  \[u^{n+1} := \sum_{i=1}^{2} E_{i}(\chi_{i} u_{i}^{n+1}).\]

● Not based on the partition of unity **Additive Schwarz (ASM)**

  \[u^{n+1} := \sum_{i=1}^{2} E_{i}(u_{i}^{n+1}).\]
Denote \( \mathcal{N} = \text{dof}(\Omega) \) and \( \mathcal{N}_j = \text{dof}(\Omega_j) \). We have the restriction operators

\[
R_i : \mathbb{R}^{\# \mathcal{N}} \mapsto \mathbb{R}^{\# \mathcal{N}_i}
\]

and the transpose is a prolongation operator

\[
R_i^T : \mathbb{R}^{\# \mathcal{N}_i} \mapsto \mathbb{R}^{\# \mathcal{N}}
\]

The local Dirichlet matrices are given by

\[
A_i := R_i A R_i^T.
\]

The partition of unity defined by matrices \( D_i \)

\[
D_i : \mathbb{R}^{\# \mathcal{N}_i} \mapsto \mathbb{R}^{\# \mathcal{N}_i}
\]

so that we have:

\[
\sum_{i=1}^{N} R_i^T D_i R_i = \text{Id}
\]
Definition: RAS (Restricted Additive Schwarz)

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

(1)

so that the iterative RAS algorithm reads:

\[
U^{n+1} = U^n + M_{\text{RAS}}^{-1} r^n, \quad r^n := F - A U^n.
\]

Definition: ASM (Additive Schwarz Method)

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

(2)

so that the iterative ASM algorithm reads:

\[
U^{n+1} = U^n + M_{\text{ASM}}^{-1} r^n.
\]
ASM and RAS in iterative version are preconditioned fixed point iterations ⇒ use Krylov methods instead.

- RAS (in conjunction with BiCGStab or GMRES) to solve

\[
M_{RAS}^{-1} AU = M_{RAS}^{-1} F.
\]

- ASM (in a CG methods)

\[
M_{ASM}^{-1} AU = M_{ASM}^{-1} F.
\]
How to evaluate the efficiency of a domain decomposition?

**Weak scalability – definition**

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

**It is not achieved with the one level method**

<table>
<thead>
<tr>
<th>Number of subdomains</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>18</td>
<td>35</td>
<td>66</td>
<td>128</td>
</tr>
</tbody>
</table>

The iteration number increases linearly with the number of subdomains in one direction.
Plateaus appear in the convergence of the Krylov methods.
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 Nabben and Vuik’s papers in SIAM J. Sci. Comp, 200X).
Adding a coarse space

We add a coarse space correction (aka second level)
Let $V_H$ be the coarse space and $Z$ be a basis, $V_H = \text{span } Z$, 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 is to use the kernel of the operator as a coarse space, this is the constant vectors, in local form this writes:

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

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

$$\sum_{i=1}^{N} R_i^T D_i R_i = I.$$
**Theoretical convergence result**

**Theorem (Widlund, Dryija)**

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 $\delta$ is the size of the overlap between the subdomains and $H$ the subdomain size.

**This does indeed work very well**

<table>
<thead>
<tr>
<th>Number of subdomains</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>18</td>
<td>35</td>
<td>66</td>
<td>128</td>
</tr>
<tr>
<td>ASM + Nicolaides</td>
<td>20</td>
<td>27</td>
<td>28</td>
<td>27</td>
</tr>
</tbody>
</table>
1 Introduction

2 Coarse space for heterogeneous problems: the DtN algorithm

3 An abstract 2-level Schwarz: the GenEO algorithm

4 Bibliography
Darcy equation with heterogeneities

\[-\nabla \cdot (\alpha(x, y) \nabla u) = 0 \text{ in } \Omega \subset \mathbb{R}^2,\]
\[u = 0 \text{ on } \partial\Omega_D,\]
\[\frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega_N.\]

Decomposition

<table>
<thead>
<tr>
<th>Jump</th>
<th>1</th>
<th>10</th>
<th>10^2</th>
<th>10^3</th>
<th>10^4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>39</td>
<td>45</td>
<td>60</td>
<td>72</td>
<td>73</td>
</tr>
<tr>
<td>ASM + Nicolaides</td>
<td>30</td>
<td>36</td>
<td>50</td>
<td>61</td>
<td>65</td>
</tr>
</tbody>
</table>
Objectives

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 does not depend on the partition nor the jumps in coefficients)
Heuristic approach: what functions should be in $\mathbb{Z}_2$?

The error satisfies the Schwarz algorithm, it is harmonic, so it satisfies a maximum principle.

$$\Omega_{i-1} e_i^{n+1} \Omega_i e_i \Omega_{i+1}$$

Fast convergence

$$\Omega_{i-1} e_i^{n+1} \Omega_i e_i \Omega_{i+1}$$

Slow convergence

**Idea**

Ensure that the error decreases quickly on the subdomain boundaries which translates to making $\frac{\partial e}{\partial n_i} \mid_{\Gamma_i}$ big.
Using the DtN operator

The **Dirichlet to Neumann operator** is defined as follows: Let $g : \Gamma_i \mapsto \mathbb{R}$,

$$
\text{DtN}_{\Omega_i}(g) = \alpha \frac{\partial v}{\partial n_i} \bigg|_{\Gamma_i},
$$

where $v$ satisfies

$$
\begin{cases}
(\text{div}(\alpha \nabla)) v = 0, & \text{in } \Omega_i, \\
v = g, & \text{on } \partial \Omega_i.
\end{cases}
$$

To construct the coarse space, we use the low frequency modes associated with the DtN operator:

$$
\text{DtN}_{\Omega_i}(v_i^\lambda) = \lambda \alpha v_i^\lambda
$$

with $\lambda$ small. The functions $v_i^\lambda$ are extended harmonically to the subdomains.
Suppose we have $(\nu_{i}^{\lambda_{k}}, \lambda_{i}^{k})_{1 \leq k \leq n_{i}}$, the eigenpairs of the local DtN maps $(\lambda_{i}^{1} \leq \lambda_{i}^{2} \leq \ldots)$ and that we have selected $m_{i}$ in each subdomain. Then let $Z$ be the coarse space built via the local DtN maps:

$$Z := (R_{i}^{T} D_{i} \tilde{V}_{i}^{\lambda_{i}^{k}})_{1 \leq i \leq N; 1 \leq k \leq m_{i}}$$

Theorem (Dolean, N., Scheichl and Spillane 2010)

Under the monotonicity of $\alpha$ in the overlapping regions:

$$\kappa\left(M_{ASM,2}^{-1} A\right) \leq C(1 + \max_{1 \leq i \leq N} \frac{1}{\delta_{i} \lambda_{i}^{m_{i}+1}})$$

where $\delta_{i}$ is the size of the overlap of domain $\Omega_{i}$ and $C$ is independent of the jumps of $\alpha$.

If $m_{i}$ is chosen so that, $\lambda_{i}^{m_{i}+1} \geq 1/H_{i}$ the convergence rate will be analogous to the constant coefficient case.
Results with the new DtN method

<table>
<thead>
<tr>
<th>Jump</th>
<th>1</th>
<th>10</th>
<th>$10^2$</th>
<th>$10^3$</th>
<th>$10^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>39</td>
<td>45</td>
<td>60</td>
<td>72</td>
<td>73</td>
</tr>
<tr>
<td>ASM + Nicolaides</td>
<td>30</td>
<td>36</td>
<td>50</td>
<td>61</td>
<td>65</td>
</tr>
<tr>
<td>ASM + DtN</td>
<td>31</td>
<td>35</td>
<td>36</td>
<td>36</td>
<td>36</td>
</tr>
</tbody>
</table>

Decomposition $\alpha(x, y)$

With DtN the jumps do not affect convergence
We put at most two modes per subdomain in the coarse space (using the automatic selection process)
Numerical results

Using FreeFEM++ [http://www.freefem.org/ff++]

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

ASM convergence for channels and inclusions – $4 \times 4$ Metis partitioning
Numerical results – Optimality

<table>
<thead>
<tr>
<th>#Z per subd.</th>
<th>ASM</th>
<th>ASM+Z\textsubscript{Nico}</th>
<th>ASM+Z\textsubscript{D2N}</th>
</tr>
</thead>
<tbody>
<tr>
<td>max(m\textsubscript{i} − 1, 1)</td>
<td></td>
<td>273</td>
<td></td>
</tr>
<tr>
<td>m\textsubscript{i}</td>
<td>614</td>
<td>543</td>
<td>36</td>
</tr>
<tr>
<td>m\textsubscript{i} + 1</td>
<td></td>
<td></td>
<td>32</td>
</tr>
</tbody>
</table>

m\textsubscript{i} is given automatically by the strategy.

- Taking one fewer eigenvalue has a huge influence on the iteration count
- Taking one more has only a small influence
PhD of Pierre Jolivet.
Since version 1.16, bundled with the Message Parsing Interface. FreeFem++ is working on the following parallel architectures (among others):

<table>
<thead>
<tr>
<th></th>
<th>N° of cores</th>
<th>Memory</th>
<th>Peak perf</th>
<th>Compilers</th>
</tr>
</thead>
<tbody>
<tr>
<td>hpc1@LJLL</td>
<td>64@2.00 Ghz</td>
<td>252 Go</td>
<td>&lt; 1 TFLOP/s</td>
<td>Intel</td>
</tr>
<tr>
<td>titane@CEA</td>
<td>121921@2.93 Ghz</td>
<td>37 To</td>
<td>140 TFLOP/s</td>
<td>Intel</td>
</tr>
<tr>
<td>babel@IDRIS</td>
<td>40960@850 Mhz</td>
<td>20 To</td>
<td>139 TFLOP/s</td>
<td>IBM+GNU</td>
</tr>
</tbody>
</table>

* + 46080 CUDA cores

http://www.idris.fr, Orsay, France.
### Strong scalability in two dimensions

**Introduction**

**Two-level methods**

**Numerical experiments**

**Conclusion**

**Computing resources**

**Implementation framework**

**Scalability tests**

Strong scalability in two dimensions

\[ p \times \sum_{i=1}^{N} \nu_i \]

<table>
<thead>
<tr>
<th>( p )</th>
<th>( T )</th>
<th>( \sum_{i=1}^{N} \nu_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>65.7 s</td>
<td>1,890</td>
</tr>
<tr>
<td>96</td>
<td>30.2 s</td>
<td>2,850</td>
</tr>
<tr>
<td>128</td>
<td>21.4 s</td>
<td>3,810</td>
</tr>
<tr>
<td>160</td>
<td>16.6 s</td>
<td>4,770</td>
</tr>
<tr>
<td>192</td>
<td>12.7 s</td>
<td>5,730</td>
</tr>
<tr>
<td>224</td>
<td>11.2 s</td>
<td>6,690</td>
</tr>
<tr>
<td>256</td>
<td>9.4 s</td>
<td>7,650</td>
</tr>
<tr>
<td>288</td>
<td>9.3 s</td>
<td>8,610</td>
</tr>
<tr>
<td>320</td>
<td>8.4 s</td>
<td>9,570</td>
</tr>
<tr>
<td>416</td>
<td>6.4 s</td>
<td>12,450</td>
</tr>
<tr>
<td>512</td>
<td>6.0 s</td>
<td>15,330</td>
</tr>
</tbody>
</table>

\[
\approx 37M \text{ unknowns}
\]

### Speed-up for a 2D problem

![Graph showing speed-up for a 2D problem](image-url)
### Strong scalability in three dimensions

**Introduction**

**Two-level methods**

**Numerical experiments**

**Conclusion**

**Computing resources**

**Implementation framework**

**Scalability tests**

\[ p_T \approx N \sum_{i=1}^{N} \nu_i \]

<table>
<thead>
<tr>
<th>( p )</th>
<th>( T ) (s)</th>
<th>( \sum_{i=1}^{N} \nu_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>96</td>
<td>26.5</td>
<td>1,920</td>
</tr>
<tr>
<td>128</td>
<td>18.6</td>
<td>2,560</td>
</tr>
<tr>
<td>160</td>
<td>13.0</td>
<td>3,200</td>
</tr>
<tr>
<td>192</td>
<td>10.4</td>
<td>3,840</td>
</tr>
<tr>
<td>224</td>
<td>8.1</td>
<td>4,480</td>
</tr>
<tr>
<td>288</td>
<td>6.9</td>
<td>5,760</td>
</tr>
</tbody>
</table>

**Approximately 8M unknowns**

\[ \approx 8M \text{ unknowns} \]

**Linear**

**Regression**

**Speed-up for a 3D problem**
Some “even bigger” problem

On *babel*, allowable memory space per core: 512MB!

4096-way decomposition:
- in $\mathbb{R}^2$, 168M unknowns,
- in $\mathbb{R}^3$, 86M unknowns.

All systems are solved with:
- coarse spaces of size $[100; 20000]$,
- less than 25 iterations.
Some “even bigger” problem

On babel, allowable memory space per core: 512MB!

4096-way decomposition:
- in $\mathbb{R}^2$, 168M unknowns,
- in $\mathbb{R}^3$, 86M unknowns.

All systems are solved with:
- coarse spaces of size $[100; 20000]$,
- less than 25 iterations.
Some “even bigger” problem

On *babel*, allowable memory space per core: 512MB!

4096-way decomposition:
- in $\mathbb{R}^2$, 168M unknowns,
- in $\mathbb{R}^3$, 86M unknowns.

All systems are solved with:
- coarse spaces of size $[100; 20000]$,
- less than 25 iterations.
1 Introduction
2 Coarse space for heterogeneous problems: the DtN algorithm
3 An abstract 2-level Schwarz: the GenEO algorithm
4 Bibliography
Main disadvantages: extension to other equations/systems such as elasticity

- Not a very natural definition of $DtN$ type operators
- Theoretical convergence proof based on Poincaré type inequalities which are not clear for other systems.

$\Rightarrow$ need to re-think the strategy of building the coarse space.
Relation of GENEO to other methods

Graham, Lechner & Scheichl
Scheichl & Vainikko
Overlapping Schwarz
MS coarse space

Galvis & Efendiev
Overlapping Schwarz
a) std. coarse space, WPI
b) gen.EVP $K_K = \lambda M_K$

Efendiev, Galvis, Lazarov & Willems
Overlapping Schwarz, abstract SPD problems
gen.EVP $K_K = \lambda K^\xi_K$

Pechstein & Scheichl
FETI (2 papers)
boundary layers
WPI for some patterns

Dolean, Nataf, Spillane & Xiang / & Scheichl
Overlapping Schwarz
gen.EVP $S_K = \lambda M^\Gamma_K$

THIS TALK
Problem setting – I

Given \( f \in (V^h)^* \) find \( u \in V^h \)

\[
a(u, v) = \langle f, v \rangle \quad \forall v \in V^h
\]

\[\iff\]

\[Au = f\]

Assumption throughout: \( A \) symmetric positive definite (SPD)

Examples:

- **Darcy**
  \[
a(u, v) = \int_\Omega \kappa \nabla u \cdot \nabla v \, dx
\]

- **Elasticity**
  \[
a(u, v) = \int_\Omega C \varepsilon(u) : \varepsilon(v) \, dx
\]

- **Eddy current**
  \[
a(u, v) = \int_\Omega \nu \text{curl } u \cdot \text{curl } v + \sigma u \cdot v \, dx
\]

Heterogeneities / high contrast in parameters
1. \( \mathcal{V}^h \) – FE space of functions in \( \Omega \) based on mesh \( \mathcal{T}^h = \{ \tau \} \)

2. \( \{ \phi_k \}_{k=1}^n \) (FE) basis of \( \mathcal{V}^h \)

3. Technical assumptions fulfilled by standard FE and bilinear forms \( a(\cdot, \cdot) \)
Overlapping decomposition: \( \Omega = \bigcup_{j=1}^{N} \Omega_j \) (\( \Omega_j \) union of elements)

\[ V_j := \text{span}\{\phi_k : \text{supp}(\phi_k) \subset \overline{\Omega_j}\} \]

such that every \( \phi_k \) is contained in one of those spaces, i.e.

\[ V^h = \sum_{j=1}^{N} V_j \]

**Example:** adding “layers” to non-overlapping partition

(partition and adding layers based on matrix information only!)
Schwarz setting – II

Local subspaces:

\[ V_j \subset V^h \quad j = 1, \ldots, N \]

**Coarse space** (defined later):

\[ V_0 \subset V^h \]

**Additive Schwarz preconditioner:**

\[
M^{-1}_{AS,2} = \sum_{j=0}^{N} R_j^\top A_j^{-1} R_j
\]

where \( A_j = R_j^\top A R_j \)

and \( R_j^\top \leftrightarrow R_j^\top : V_j \rightarrow V^h \) natural embedding
Overlapping zone: \( \Omega_j \circ = \{ x \in \Omega_j : \exists i \neq j : x \in \Omega_i \} \)

Observation: \( \Xi_j | \Omega_j \setminus \Omega_j \circ = \text{id} \)

**Coarse space** should be a sum of local contributions:

\[
V_0 = \sum_{j=1}^{N} V_{0,j} \quad \text{where} \quad V_{0,j} \subset V_j
\]

E.g. \( V_{0,j} = \text{span}\{\Xi_j \rho_{j,k}\}_{k=1}^{m_j} \)
Choice of coarse space (continued)

ASM theory needs **stable splitting**:

\[ \mathbf{v} = \mathbf{v}_0 + \sum_{j=1}^{N} \mathbf{v}_j \]

Suppose \( \mathbf{v}_0 = \sum_{j=1}^{N} \Xi_j \Pi_j \mathbf{v}|_{\Omega_j} \) where \( \Pi_j \ldots \) local projector

\[
| \Xi_j (\mathbf{v} - \Pi_j \mathbf{v}) |_{a,\Omega_j}^2 = | \Xi_j (\mathbf{v} - \Pi_j \mathbf{v}) |_{a,\Omega_j^o}^2 + | \Xi_j (\mathbf{v} - \Pi_j \mathbf{v}) |_{a,\Omega_j \setminus \Omega_j^o}^2
\]

**HOW?**

\[
\leq C | \mathbf{v} |_{a,\Omega_j}^2
\]

("Minimal" requirements:
- \( \Pi_j \) be \( a \)-orthogonal
- Stability estimate: \( | \Xi_j (\mathbf{v} - \Pi_j \mathbf{v}) |_{a,\Omega_j^o}^2 \leq c | \mathbf{v} |_{a,\Omega_j}^2 \)

\((a,D \) denotes the restriction of \( a \) to \( D \))
ASM theory needs **stable splitting**:

\[ \mathbf{v} = \mathbf{v}_0 + \sum_{j=1}^{N} \mathbf{v}_j \]

Suppose \( \mathbf{v}_0 = \sum_{j=1}^{N} \Xi_j \Pi_j \mathbf{v}_{|\Omega_j} \) where \( \Pi_j \) are local projectors

\[
\underbrace{\Xi_j (\mathbf{v} - \Pi_j \mathbf{v})}_{V_j} \bigg|_{a,\Omega_j}^2 = \| \Xi_j (\mathbf{v} - \Pi_j \mathbf{v}) \|_{a,\Omega_j}^2 + \| \Xi_j (\mathbf{v} - \Pi_j \mathbf{v}) \|_{a,\Omega_j \setminus \Omega_j}^2
\]

**HOW?**

\[ \leq C \| \mathbf{v} \|_{a,\Omega_j}^2 \]

("Minimal" requirements:

- \( \Pi_j \) be \( a \)-orthogonal
- Stability estimate: \( \| \Xi_j (\mathbf{v} - \Pi_j \mathbf{v}) \|_{a,\Omega_j}^2 \leq c \| \mathbf{v} \|_{a,\Omega_j}^2 \)

\((a,D) \) denotes the restriction of \( a \) to \( D \)
“Minimal” requirements:

1. $\Pi_j$ be $a$-orthogonal
2. Stability estimate: $|\Xi_j(v - \Pi_j v)|^2_{a, \Omega_j} \leq c |v|^2_{a, \Omega_j}$

Fulfillment of 2

If there exists a non-zero function $w$ such that $|w|_{a, \Omega_j} = 0$, it is necessary to project on $\text{Span}(w)$.

The kernel of a Darcy equation is the constant function and that of elasticity is spanned by rigid body motions.

The corresponding coarse space will be referred to as ZEM (zero energy modes).

For highly heterogeneous problems, we take a larger coarse space deduced from the stability estimate.
Abstract eigenvalue problem

**Gen.EVP** per subdomain:

Find \( p_{j,k} \in V_{h|\Omega_j} \) and \( \lambda_{j,k} \geq 0 \):

\[
a_{\Omega_j}(p_{j,k}, \nu) = \lambda_{j,k} a_{\Omega_j^o}(\Xi_j p_{j,k}, \Xi_j \nu) \quad \forall \nu \in V_{h|\Omega_j}
\]

\[
A_j p_{j,k} = \lambda_{j,k} X_j A_j^o X_j p_{j,k} \quad (X_j \ldots \text{diagonal})
\]

\( a_D \ldots \) restriction of \( a \) to \( D \)

---

In the two-level ASM:
Choose first \( m_j \) eigenvectors per subdomain:

\[
V_0 = \text{span}\{\Xi_j p_{j,k}\}_{j=1,\ldots,N}^{k=1,\ldots,m_j}
\]

This automatically includes Zero Energy Modes.
Abstract eigenvalue problem

**Gen.EVP** per subdomain:

Find \( p_{j,k} \in V_{h|\Omega_j} \) and \( \lambda_{j,k} \geq 0 \):

\[
a_{\Omega_j}(p_{j,k}, v) = \lambda_{j,k} a_{\Omega_j^\circ}(\Xi_j p_{j,k}, \Xi_j v) \quad \forall v \in V_{h|\Omega_j}
\]

\[
A_j p_{j,k} = \lambda_{j,k} X_j A_j^\circ X_j p_{j,k} \quad (X_j \ldots \text{diagonal})
\]

\( a_D \ldots \) restriction of \( a \) to \( D \)

---

**In the two-level ASM:**

Choose first \( m_j \) eigenvectors per subdomain:

\[
V_0 = \text{span}\{\Xi_j p_{j,k}\}_{j=1,\ldots,N}^{k=1,\ldots,m_j}
\]

This automatically includes Zero Energy Modes.
Comparison with existing works

Galvis & Efendiev (SIAM 2010):
\[ \int_{\Omega_j} \kappa \nabla p_{j,k} \cdot \nabla v \, dx = \lambda_{j,k} \int_{\Omega_j} \kappa p_{j,k} v \, dx \quad \forall v \in V_{h|\Omega_j} \]

Efendiev, Galvis, Lazarov & Willems (submitted):
\[ a_{\Omega_j}(p_{j,k}, v) = \lambda_{j,k} \sum_{i \in \text{neighb}(j)} a_{\Omega_j}(\xi_j \xi_i p_{j,k}, \xi_j \xi_i v) \quad \forall v \in V_{|\Omega_j} \]
\[ \xi_j \ldots \text{partition of unity, calculated adaptively (MS)} \]

Our gen.EVP:
\[ a_{\Omega_j}(p_{j,k}, v) = \lambda_{j,k} a_{\Omega_j}(\Xi_j p_{j,k}, \Xi_j v) \quad \forall v \in V_{h|\Omega_j} \]

both matrices typically singular \( \implies \lambda_{j,k} \in [0, \infty] \)
Two technical assumptions.

Theorem (Spillane, Dolean, Hauret, N., Pechstein, Scheichl)

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

$$
\kappa(M^{-1}_{ASM,2}A) \leq (1 + k_0) \left[ 2 + k_0 (2k_0 + 1) \max_{j=1}^{N} \left( 1 + \frac{1}{\lambda_{j,m_j+1}} \right) \right]
$$

Possible criterion for picking $m_j$: (used in our Numerics)

$$
\lambda_{j,m_j+1} < \frac{\delta_j}{H_j}
$$

$H_j \ldots$ subdomain diameter, $\delta_j \ldots$ overlap
### Domain & Partitions

**Iterations (CG) vs. jumps**  
**Code: Matlab & FreeFem++**

<table>
<thead>
<tr>
<th>$\kappa_2$</th>
<th>AS-1</th>
<th>AS-ZEM</th>
<th>$\text{dim}(V_H)$</th>
<th>GENEO</th>
<th>$\text{dim}(V_H)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>16</td>
<td>(8)</td>
<td>16</td>
<td>(8)</td>
</tr>
<tr>
<td>$10^2$</td>
<td>31</td>
<td>24</td>
<td>(8)</td>
<td>17</td>
<td>(15)</td>
</tr>
<tr>
<td>$10^4$</td>
<td>37</td>
<td>30</td>
<td>(8)</td>
<td>21</td>
<td>(15)</td>
</tr>
<tr>
<td>$10^6$</td>
<td>36</td>
<td>29</td>
<td>(8)</td>
<td>18</td>
<td>(15)</td>
</tr>
</tbody>
</table>

AS-1: 1-level ASM  
AS-ZEM: $m_j = 1$  
GENEO: $\lambda_j, m_{j+1} < \delta_j/H_j$
# Iterations (CG) vs. number of subdomains

## regular partition

<table>
<thead>
<tr>
<th>subd.</th>
<th>dofs</th>
<th>AS-1</th>
<th>AS-ZEM</th>
<th>dim($V_H$)</th>
<th>GENE0</th>
<th>dim($V_H$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4840</td>
<td>14</td>
<td>15</td>
<td>(4)</td>
<td>10</td>
<td>(6)</td>
</tr>
<tr>
<td>8</td>
<td>9680</td>
<td>26</td>
<td>22</td>
<td>(8)</td>
<td>11</td>
<td>(14)</td>
</tr>
<tr>
<td>16</td>
<td>19360</td>
<td>51</td>
<td>36</td>
<td>(16)</td>
<td>13</td>
<td>(30)</td>
</tr>
<tr>
<td>32</td>
<td>38720</td>
<td>&gt;100</td>
<td>61</td>
<td>(32)</td>
<td>13</td>
<td>(62)</td>
</tr>
</tbody>
</table>

## METIS partition

<table>
<thead>
<tr>
<th>subd.</th>
<th>dofs</th>
<th>AS-1</th>
<th>AS-ZEM</th>
<th>dim($V_H$)</th>
<th>GENE0</th>
<th>dim($V_H$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4840</td>
<td>21</td>
<td>18</td>
<td>(4)</td>
<td>15</td>
<td>(7)</td>
</tr>
<tr>
<td>8</td>
<td>9680</td>
<td>36</td>
<td>29</td>
<td>(8)</td>
<td>18</td>
<td>(15)</td>
</tr>
<tr>
<td>16</td>
<td>19360</td>
<td>65</td>
<td>45</td>
<td>(16)</td>
<td>22</td>
<td>(31)</td>
</tr>
<tr>
<td>32</td>
<td>38720</td>
<td>&gt;100</td>
<td>79</td>
<td>(32)</td>
<td>34</td>
<td>(63)</td>
</tr>
</tbody>
</table>
$E_1 = 2 \cdot 10^{11}$
$\nu_1 = 0.3$

$E_2 = 2 \cdot 10^7$
$\nu_2 = 0.45$

METIS partitions with 2 layers added

<table>
<thead>
<tr>
<th>subd.</th>
<th>dofs</th>
<th>AS-1</th>
<th>AS-ZEM</th>
<th>$(V_H)$</th>
<th>GENEEO</th>
<th>$(V_H)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>13122</td>
<td>93</td>
<td>134</td>
<td>(12)</td>
<td>42</td>
<td>(42)</td>
</tr>
<tr>
<td>16</td>
<td>13122</td>
<td>164</td>
<td>165</td>
<td>(48)</td>
<td>45</td>
<td>(159)</td>
</tr>
<tr>
<td>25</td>
<td>13122</td>
<td>211</td>
<td>229</td>
<td>(75)</td>
<td>47</td>
<td>(238)</td>
</tr>
<tr>
<td>64</td>
<td>13122</td>
<td>279</td>
<td>167</td>
<td>(192)</td>
<td>45</td>
<td>(519)</td>
</tr>
</tbody>
</table>
Elasticity

**Iterations (CG) vs. number of subdomains**

\[ E_1 = 2 \cdot 10^{11} \]
\[ \nu_1 = 0.3 \]
\[ E_2 = 2 \cdot 10^7 \]
\[ \nu_2 = 0.45 \]

Relative error vs. iterations
16 regular subdomains

<table>
<thead>
<tr>
<th>subd.</th>
<th>dofs</th>
<th>AS-1</th>
<th>AS-ZEM</th>
<th>((V_H))</th>
<th>GENELO</th>
<th>((V_H))</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1452</td>
<td>79</td>
<td>54</td>
<td>(24)</td>
<td>16</td>
<td>(46)</td>
</tr>
<tr>
<td>8</td>
<td>29040</td>
<td>177</td>
<td>87</td>
<td>(48)</td>
<td>16</td>
<td>(102)</td>
</tr>
<tr>
<td>16</td>
<td>58080</td>
<td>378</td>
<td>145</td>
<td>(96)</td>
<td>16</td>
<td>(214)</td>
</tr>
</tbody>
</table>

AS-ZEM (Rigid body motions): \(m_j = 6\)
Remarks:

- Implementation requires only element stiffness matrices + connectivity
- Proof works for any partition of unity
  (changes the eigenproblem and coarse space)

Outlook:

- More testing & comparison to other methods
- Solution of the Eigenproblems (LAPACK $\mapsto$ LOBPCG)
- Coarse space dimension reduction?
- Coarse problem satisfies assembling property
  $\mapsto$ multilevel method $\mapsto$ link to $\sigma$AMGe ?
- More applications within a FreeFem++ MPI implementation
- Other discretizations: finite volume, finite difference, ...
Remarks:
- Implementation requires only element stiffness matrices + connectivity
- Proof works for any partition of unity
  (changes the eigenproblem and coarse space)

Outlook:
- More testing & comparison to other methods
- Solution of the Eigenproblems (LAPACK $\mapsto$ LOBPCG)
- Coarse space dimension reduction?
- Coarse problem satisfies assembling property
  $\mapsto$ multilevel method — link to $\sigma$AMGe?
- More applications within a FreeFem++ MPI implementation
- Other discretizations: finite volume, finite difference, ...
Outline

1 Introduction

2 Coarse space for heterogeneous problems: the DtN algorithm

3 An abstract 2-level Schwarz: the GenEO algorithm

4 Bibliography
Bibliography Schwarz: DtN and GenEO

Preprints available on HAL:
http://hal.archives-ouvertes.fr/


THANK YOU FOR YOUR ATTENTION!
Preprints available on HAL: http://hal.archives-ouvertes.fr/


THANK YOU FOR YOUR ATTENTION!