Two-Level Domain Decomposition Methods for Highly Heterogeneous Darcy Equations. Connections with Multiscale Methods

Victorita Dolean, Pierre Jolivet, Frédéric Nataf, Nicole Spillane, Hua Xiang

Photos: DOI: 10.1051/ogst/2013204, IFPEN, X.

Dossier

This paper is a part of the hereunder thematic dossier published in OGST Journal, Vol. 69, No. 4, pp. 507-766 and available online here

Cet article fait partie du dossier thématique ci-dessous publié dans la revue OGST, Vol. 69, n°4 pp. 507-766 et téléchargeable ici

DOSSIER Edited by/Sous la direction de : **Z. Benjelloun-Touimi**

Geosciences Numerical Methods
Modélisation numérique en géosciences

Oil & Gas Science and Technology – Rev. IFP Energies nouvelles, Vol. 69 (2014), No. 4, pp. 507-766

IFP Energies nouvelles

Oil & Gas Science and Technology – Rev. IFP Energies nouvelles, Vol. 69 (2014), No. 4, pp. 731-752

DOI: 10.2516/ogst/2013206

Dossier

Geosciences Numerical Methods
Modélisation numérique en géosciences

Two-Level Domain Decomposition Methods for Highly Heterogeneous Darcy Equations. Connections with Multiscale Methods

Victorita Dolean[1], Pierre Jolivet[2], Frédéric Nataf[2]*, Nicole Spillane[2] and Hua Xiang[3]

[1] *University of Strathclyde, Department of Mathematics and Statistics, 26 Richmond Street, Glasgow G1 1XH, Scotland - UK*
[2] *Laboratoire J.L. Lions, CNRS UMR 7598, Université Pierre et Marie Curie, Équipe Alpines, INRIA, 4 place Jussieu, 75252, Paris Cedex 05 - France*
[3] *School of Mathematics and Statistics, Wuhan University, 430072 Wuhan - P. R. China*
e-mail: Victorita.Dolean@strath.ac.uk - jolivet@ann.jussieu.fr - nataf@ann.jussieu.fr - spillane@ann.jussieu.fr - hxiang@whu.edu.cn

* Corresponding author

Résumé — **Méthodes de décomposition de domaine à deux niveaux pour les équations de Darcy à coefficients très hétérogènes. Liens avec les méthodes multi-échelles** — Les écoulements multiphasiques en milieux poreux conduisent à la solution de systèmes d'Équations aux Dérivées Partielles (EDP) à coefficients très hétérogènes. Nous nous concentrons sur les méthodes de décomposition de domaine avec recouvrement de type Schwarz sur calculateurs parallèles et sur les méthodes multi-échelles. Nous présentons un espace grossier [Nataf F., Xiang H., Dolean V., Spillane N. (2011) *SIAM J. Sci. Comput*. **33**, 4, 1623-1642] qui est robuste, même en présence de telles hétérogénéités. L'approche méthodes de décomposition de domaine à deux niveaux est comparée aux méthodes multi-échelles.

Abstract — ***Two-Level Domain Decomposition Methods for Highly Heterogeneous Darcy Equations. Connections with Multiscale Methods*** — *Multiphase, compositional porous media flow models lead to the solution of highly heterogeneous systems of Partial Differential Equations (PDE). We focus on overlapping Schwarz type methods on parallel computers and on multiscale methods. We present a coarse space [Nataf F., Xiang H., Dolean V., Spillane N. (2011)* SIAM J. Sci. Comput. ***33****, 4, 1623-1642] that is robust even when there are such heterogeneities. The two-level domain decomposition approach is compared to multiscale methods.*

NOMENCLATURE

ASM	Additive Schwarz Method
BC	Boundary Conditions
BICGSTAB	BIConjugate Gradiant STABilized
CG	Conjugate Gradient
DD	Domain Decomposition
d.o.f.	Degrees of freedom
DtN	Dirichlet-to-Neumann
GMRES	Generalized Minimal RESidual
JSM	Jacobi Schwarz Method
MsFEM	Multiscale Finite Element Method
MsFV	Multiscale Finite Volume
PDE	Partial Differential Equations
RAS	Restricted Additive Schwarz
SPD	Symmetric Positive Definite

INTRODUCTION

Multiphase, compositional porous media flow models, used in reservoir simulations or basin modeling, lead to the solution of complex non linear systems of Partial Differential Equations (PDE). These PDE are typically discretized using a cell-centered finite volume scheme and a fully implicit Euler integration in time in order to allow for large time steps. After Newton type linearization, one ends up with the solution of a linear system at each Newton iteration which represents up to 90 percents of the total simulation elapsed time. The corresponding pressure block matrix is related to the discretization of a Darcy equation with high contrasts and anisotropy in the coefficients. We focus on overlapping Schwarz type methods on parallel computers and on multiscale methods.

Coarse spaces are instrumental in obtaining scalability for domain decomposition methods. For matrices arising from problems with smooth coefficients, it is possible to build efficient coarse spaces based on domain wise constant vectors [1]. For problems with high heterogeneities, these simple coarse spaces do not work well. Here, we present a recent coarse space [2] that is robust even when there are such heterogeneities. We achieve this by solving local generalized eigenvalue problems which isolate the terms responsible for slow convergence. Building efficient coarse spaces is closely related to multiscale methods which also aim to reduce the computational cost of large scale problems.

An outline of the paper is as follows. Section 1 consists in an introduction to one level Schwarz methods. Material is basic but the presentation is quite new. In Section 2, we present a recent spectral coarse space which adapts automatically to the heterogeneities of the problem. In Section 3, we present results of large scale computations. In Section 4, it is compared to multiscale methods. Then, we conclude and present prospects on adaptation of the spectral coarse space to finite volume discretizations.

1 SCHWARZ METHODS

We start with the original Schwarz algorithm [3] at the continuous (*i.e.* partial differential equations) level whose parallel version is named Jacobi-Schwarz Method (JSM). We introduce two variants that are at the origin of the popular Additive Schwarz Method (ASM) and Restricted Additive Schwarz (RAS [4]) algorithms. The first one has been the subject of hundreds of papers (see [1] and references therein). The second one is the default parallel solver of the parallel package software PETSc [5]. This presentation shows in a unified setting the connections between these three algorithms.

1.1 Three Schwarz Algorithms at the Continuous Level

Hermann Schwarz was a German analyst of the 19th century. He was interested in proving existence and uniqueness of the Poisson problem. At his time, there were no Sobolev spaces nor Lax-Milgram theorem. The only available tool was the Fourier transform, limited by its very nature to simple geometries. In order to consider more general situations, Schwarz devised an algorithm based on solving iteratively Poisson problem set on a union of simple geometries. Let the domain Ω be the union of a disk and a rectangle, see Figure 1 and consider the Poisson problem:

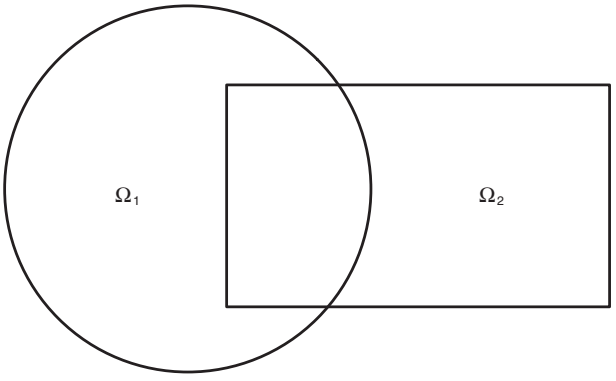

Figure 1

A complex domain Ω made from the union of two simple geometries.

Find $u : \Omega \to \mathbb{R}$ such that:

$$\begin{aligned} -\Delta u &= f \quad \text{in} \quad \Omega \\ u &= 0 \quad \text{on} \quad \partial\Omega \end{aligned} \tag{1}$$

The Schwarz algorithm is an iterative method based on solving alternatively subproblems in domains Ω_1 and Ω_2. It updates $\left(u_1^n, u_2^n\right) \to \left(u_1^{n+1}, u_2^{n+1}\right)$ by:

$$\begin{aligned} -\Delta\left(u_1^{n+1}\right) &= f \quad \text{in} \quad \Omega_1 \\ u_1^{n+1} &= 0 \quad \text{on} \quad \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} &= u_2^n \quad \text{on} \quad \partial\Omega_1 \cap \bar{\Omega}_2 \end{aligned} \tag{2a}$$

then,

$$\begin{aligned} -\Delta\left(u_2^{n+1}\right) &= f \quad \text{in} \quad \Omega_2 \\ u_2^{n+1} &= 0 \quad \text{on} \quad \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} &= u_1^{n+1} \quad \text{on} \quad \partial\Omega_2 \cap \bar{\Omega}_1 \end{aligned} \tag{2b}$$

Schwarz proved the convergence of the algorithm and thus the well-posedness of the Poisson problem in complex geometries.

With the advent of digital computers, this method also acquired a practical interest as an iterative linear solver. Subsequently, parallel computers became available and a small modification of the algorithm makes it suited to these architectures. It is sufficient to solve concurrently in all subdomains, $i = 1, 2$:

$$\begin{aligned} -\Delta\left(u_i^{n+1}\right) &= f \quad \text{in} \quad \Omega_i \\ u_i^{n+1} &= 0 \quad \text{on} \quad \partial\Omega_i \cap \partial\Omega \\ u_i^{n+1} &= u_{3-i}^n \quad \text{on} \quad \partial\Omega_i \cap \bar{\Omega}_{3-i} \end{aligned} \tag{3}$$

It is easy to see that if the algorithm converges, the solutions u_i^∞, $i = 1, 2$ in the intersection of the subdomains take the same values. Indeed, in the overlap $\Omega_{12} := \Omega_1 \cap \Omega_2$, let $e^\infty := u_1^\infty - u_2^\infty$. By the last line of (3), we know that $e^\infty = 0$ on $\partial\Omega_{12}$. By linearity of the equation, we also have that e^∞ is harmonic. Thus, e^∞ solves the homogeneous well posed BVP:

$$\begin{aligned} -\Delta(e^\infty) &= 0 \quad \text{in} \quad \Omega_{12} \\ e^\infty &= 0 \quad \text{on} \quad \partial\Omega_{12} \end{aligned}$$

and thus $e^\infty = 0$.

Algorithms (2) and (3) act on the local functions $(u_i)_{i=1,2}$. In order to write algorithms that act on global functions in $H^1(\Omega)$, the space in which problem (1) is naturally posed, we need extension operators, E_i so that 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 Ω_i. We also need 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 such that:

$$w = \sum_{i=1}^{2} E_i\left(\chi_i\, w_{|\Omega_i}\right) \tag{4}$$

for any function $w : \Omega \mapsto \mathbb{R}$. This definition of a partition of unity is closer to the computer implementation than the classical definition of a partition of unity functions.

There are two ways to write related algorithms that act on functions $u^n \in H^1(\Omega)$. The first possibility is: let u_n be an approximation to the solution to the Poisson problem (1), u^{n+1} is computed by solving first local subproblems:

$$\begin{aligned} -\Delta\left(u_i^{n+1}\right) &= f \quad \text{in} \quad \Omega_i \\ u_i^{n+1} &= 0 \quad \text{on} \quad \partial\Omega_i \cap \partial\Omega \\ u_i^{n+1} &= u^n \quad \text{on} \quad \partial\Omega_i \cap \bar{\Omega}_{3-i} \end{aligned} \tag{5}$$

and then gluing them together using the partition of unity functions:

$$u^{n+1} := \sum_{i=1}^{2} E_i\left(\chi_i\, u_i^{n+1}\right) \tag{6}$$

A second possibility consists in replacing the above formula by a simpler formula not based on the partition of unity:

$$u^{n+1} := \sum_{i=1}^{2} E_i\left(u_i^{n+1}\right) \tag{7}$$

Starting from the original Schwarz algorithm (2) that is sequential, we have thus three continuous algorithms that are essentially parallel:

- algorithm (3) Jacobi Schwarz Method (JSM);
- algorithm (5-6) Restricted Additive Schwarz (RAS);
- algorithm (5-7) Additive Schwarz Method (ASM).

These algorithms although closely related are different in nature. The JSM method acts on a pair of local functions $(u_1^n\, u_2^n)$ whereas RAS and ASM act on a global function u_n. Note that in the overlapping region, algorithms RAS and ASM update the solution in a different way. Algorithm ASM seems rather bizarre since it does not converge to the exact solution in the intersection $\Omega_1 \cap \Omega_2$. But its algebraic form given by (10) when used a preconditioner as explained in the sequel has the advantage to be Symmetric Positive Definite (SPD). On the contrary the algebraic counterpart to RAS given by (9) is unsymmetric.

1.2 Schwarz Algorithms at the Algebraic Level

So far, we have given a continuous presentation of domain decomposition methods. Actually, these methods are used in their algebraic form to solve linear systems arising from the discretization of partial differential equations. We now give the matrix counterpart of these algorithms.

For this, we first give a kind of dictionary to go from the continuous level to the discrete one:

- the counterparts of a domain Ω and of an overlapping decomposition $\Omega = \cup_{i=1}^{N} \Omega_i$ are a set of degrees of freedom (d.o.f.) $\mathcal{N}$ and a decomposition in subsets $\mathcal{N} = \cup_{i=1}^{N} \mathcal{N}_i$;
- a function $u : \Omega \to \mathbb{R}$ corresponds a vector $U \in \mathbb{R}^{\mathcal{N}}$;
- the restriction of a function $u : \Omega \to \mathbb{R}$ to a subdomain Ω_i, $1 \leq i \leq N$ is analog to the restriction $R_i\, U$ of a vector $U \in \mathbb{R}^{\mathcal{N}}$ to subset $\mathcal{N}_i$. Matrix R_i is a Boolean rectangular of size $\mathcal{N}_i \times \mathcal{N}$;
- similarly, $E_i(u_i)$ the extension by zero of a function $u_i : \Omega_i \to \mathbb{R}$ to a function $\Omega \to \mathbb{R}$ corresponds at the algebraic level to $R_i^T\, U_i$ where R_i^T is the transpose of matrix R_i and $U_i \in \mathbb{R}^{\mathcal{N}_i}$ is a local vector;
- the counterparts of partition of unity functions χ_i, $1 \leq i \leq N$ are diagonal matrices with positive entries, of size $\mathcal{N}_i \times \mathcal{N}_i$ s. t. $Id = \sum_{i=1}^{N} R_i^T\, D_i\, R_i$;
- after discretization, solving Poisson problem (1) amounts to solving a SPD linear system:

$$AU = F \tag{8}$$

- solving a local subproblem in a subdomain Ω_i such as in Equations (3) or (5) corresponds at the algebraic level to solving linear systems of the form $R_i\, A\, R_i^T\, U_i^{n+1} = F_i^n$.

We now define, at the algebraic level, the RAS and ASM algorithms and not JSM since it is seldom used and is more complex to define. As for the counterpart of the RAS algorithm (5-6), we give the following definition:

$$M_{RAS}^{-1} := \sum_{i=1}^{N} R_i^T D_i \left(R_i A R_i^T\right)^{-1} R_i \tag{9}$$

so that the iterative RAS algorithm reads:

$$U^{n+1} = U^n + M_{RAS}^{-1} r^n$$

where $r^n := F - A\, U^n$.

As for the counterpart of the ASM algorithm (5-7), we give the following definition:

$$M_{ASM}^{-1} := \sum_{i=1}^{N} R_i^T D_i \left(R_i A R_i^T\right)^{-1} R_i \tag{10}$$

so that the iterative ASM algorithm reads:

$$U^{n+1} = U^n + M_{ASM}^{-1} r^n$$

As is well known, such fixed point methods are out performed by Krylov based iterative solvers such the Conjugate Gradient (CG) algorithm of the generalized minimal residual method (GMRES), see the book by Saad [6] and references therein. In our context, using these methods amounts to solve the linear system (8) by a CG algorithm preconditioned by the symmetric preconditioner M_{ASM} or by a GMRES algorithm preconditioned by the unsymmetric preconditioner M_{RAS}. In both cases, the convergence properties are related to the spectral properties of the preconditioned operator $M_{ASM \text{ or } RAS}^{-1} A$. The RAS [7] is the default parallel solver in the PETSc package. For the ASM many theoretical results have been derived [1].

2 ADAPTIVE SPECTRAL COARSE SPACE

The domain decomposition methods presented so far were written for a two subdomain decomposition. Their extension to an arbitrary number N of subdomains $(\Omega_i)_{1 \leq i \leq N}$ is only a matter of notation. It is sufficient in definitions of the previous section to sum over all subdomains from $i = 1$ to $i = N$. But, when the number of subdomains is large, plateaus appear in the convergence of Schwarz domain decomposition methods. This is the case even for a regular problem such as the Poisson problem (1). The problem comes from the fact the preconditioner lacks of a global mechanism for exchange of information. Preconditioners RAS and ASM defined in the previous sections are called one-level methods. Data are exchanged only from one subdomain to its direct neighbors. But the solution in each subdomain depends on the right handside in all subdomains. Let us call N_d the number of subdomains in one direction. Then, for instance, the leftmost domain of Figure 2 needs at least N_d iterations before knowing something about the value of the right handside f in the rightmost subdomain. The length of the plateau is thus typically related to the number of subdomains in one direction and therefore to the notion of scalability met in the context of high performance computing.

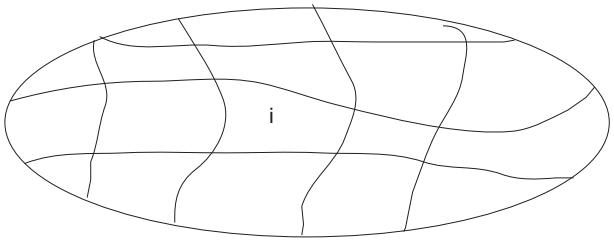

Figure 2

Decomposition into many subdomains.

In order to achieve scalability of the Domain Decomposition (DD) method, we introduce two-level domain decomposition methods *via* a coarse space correction. The precise motivation and linear algebra setting are given in Section 2.1 for a problem with smooth coefficients. A new approach Section 2.2 introduced in [2, 8] is necessary to achieve scalability for arbitrary highly heterogeneous coefficients. A condition number estimate theorem supports the approach. The method is tested in Section 2.4 on difficult heterogeneous test cases including channelized medium. In practice, the coarse space seems to be optimal, see Table 10 in Section 2.4.1.

2.1 Need for a Two-Level Method

When the number of subdomains is large, plateaus appear in the convergence of Schwarz domain decomposition methods. The remedy will consist in the introduction of a two-level preconditioner *via* a coarse space correction.

The problem and its cure are well illustrated in Figure 3 for a domain decomposition into 64 strips. The one level method has a long plateau in the convergence whereas with a coarse space correction convergence is quite fast. For instance, in Figure 4 [9], we consider a 2D problem decomposed into 2×2, 4×4 and 8×8 subdomains. For each domain decomposition, we have two curves: one with a one-level method and the second with a coarse grid correction which is denoted by M2. We see that for the one-level curves, the plateau has a size proportional to the number of subdomains in one direction. In two-level methods, a small problem of size typically the number of subdomains couples all subdomains at each iteration. It is through this mechanism that scalability can be achieved.

From a condition number point of view, stagnation corresponds to a few very low eigenvalues in the spectrum of the preconditioned problem. Using preconditioners M_{ASM} or M_{RAS}, we can remove the influence of

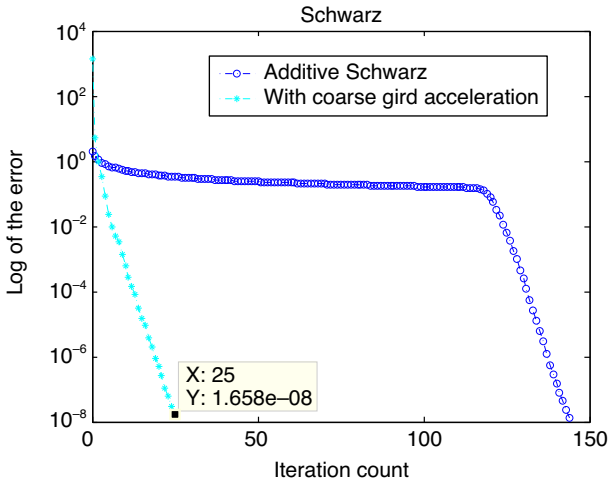

Figure 3

Convergence curves with and without a coarse space correction for a decomposition into 64 strips.

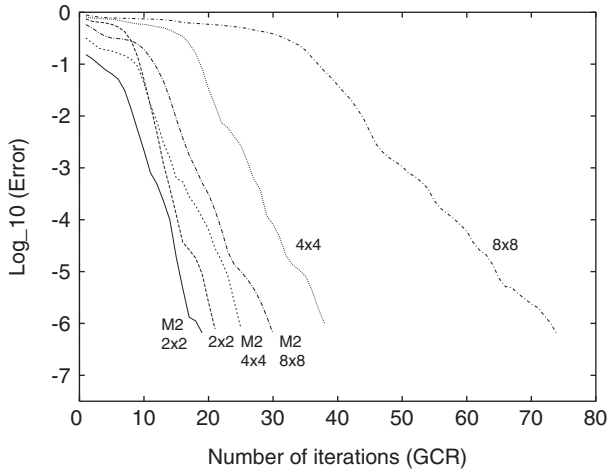

Figure 4

Japhet, Nataf and Roux [9].

very large eigenvalues of the coefficient matrix, which correspond to high frequency modes. Indeed, it has been proved that for a SPD matrix, the largest eigenvalue of the preconditioned system by M_{ASM} is bounded by the number of colors needed to color the overlapping subdomains with different colors for adjacent subdomains, see [1] or [6] for instance. But the small eigenvalues still exist and hamper the convergence. These small eigenvalues correspond to low frequency modes and represent certain global information. We need a suitable coarse grid space to efficiently deal with them.

A classical remedy consists in the introduction of a coarse grid or coarse space correction that couples all subdomains at each iteration of the iterative method. This is closely related to deflation technique classical in linear algebra, see Nabben and Vuik's paper [10]. Suppose we have identified the modes corresponding to the slow convergence of the iterative method used to solve the linear system:

$$Ax = b$$

with a preconditioner M, in our case a domain decomposition method. That is, we have some *a priori* knowledge on the small eigenvalues of the preconditioned system $M^{-1}A$. For a Poisson problem, these slow modes correspond to constant functions that are in the null space (kernel) of the Laplace operators. For a homogeneous elasticity problem, they correspond to the rigid body motions. Let us call Z the rectangular matrix whose columns correspond to these slow modes. There are algebraic ways to incorporate these informations to accelerate the domain decomposition method. We give here the presentation that is classical in domain decomposition methods. In the case where A is SPD, the starting point is to consider the minimization problem:

$$\min_{\beta} A(y + Z\beta) - b_{A^{-1}}$$

It corresponds to finding the best correction to an approximate solution y by a vector $Z\beta$ in the vector space spanned by the n_c columns of Z. This problem is equivalent to:

$$\min_{\beta \in \mathbb{R}^{n_c}} 2(Ay - b, Z\beta)_2 + (AZ\beta, AZ\beta)_2$$

and whose solution is:

$$\beta = \left(Z^T AZ\right)^{-1} Z^T (b - Ay)$$

Thus, the correction term is:

$$Z\beta = Z\left(Z^T AZ\right)^{-1} Z^T (b - Ay)$$

Let $R_0 := Z^T$ and $r = b - Ay$ be the residual associated to the approximate solution y, the best correction that belongs to the vector space spanned by the columns of Z reads:

$$R_0^T \left(R_0 A R_0^T\right)^{-1} R_0 r$$

When using such an approach with an Additive Schwarz Method (ASM), it is natural to introduce an Additive correction to the Additive Schwarz Method:

TABLE 1

Iteration counts for a Poisson problem on a domain decomposed into strips. The number of unknowns is proportional to the number of subdomains (weak scalability)

N subdomains	Schwarz	With coarse grid
4	18	25
8	37	22
16	54	24
32	84	25
64	144	25

$$\mathscr{M}_{ASM,2}^{-1} := R_0^T \left(R_0 A R_0^T\right)^{-1} R_0 + \sum_{i=1}^{N} R_i^T \left(R_i A R_i^T\right)^{-1} R_i \quad (11)$$

where the R_i's ($1 \leq i \leq N$) are the restriction operators to the overlapping subdomains. The structure of the two level preconditioner $\mathscr{M}_{ASM,2}^{-1}$ is thus the same than in the one level method. Compared to the one level Schwarz method where only local subproblems have to be solved in parallel, the two-level method adds the solution of a linear system in a sequential way with the matrix $R_0 A R_0^T$. This problem couples all subdomains at each iteration. But this matrix is a small $O(N \times N)$ square matrix and the extra cost is negligible compared to the gain. Indeed, in Table 1 we display the iteration counts for a decomposition of the domain in an increasing number of subdomains. In Figure 3, we see that without a coarse grid correction, the convergence curve of the one level Schwarz method has a very long plateau that can be bypassed by a two-level method.

We give here a precise definition to Z for a Poisson problem. This construction was introduced in Nicolaides [11]. We take Z so that it has a domain decomposition structure. Z is defined by vectors which have local support in the subdomains and so that the constant function $\mathbf{1}$ belongs to the vector space spanned by Z. Recall that we have a partition of unity in the following sense: let D_i, $1 \leq i \leq N$, be matrices:

$$D_i : \mathbb{R}^{dim(\mathscr{N}_i)} \mapsto \mathbb{R}^{dim(\mathscr{N}_i)} \quad (12)$$

so that we have:

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

We define Z such that the i-th column of Z is:

$$Z_i := R_i^T D_i R_i \mathbf{1} \text{ for } 1 \leq i \leq N \quad (13)$$

where $\mathbf{1}$ is the vector full of ones. The structure of Z is thus the following:

$$Z_{Nico} = \begin{bmatrix} D_1 R_1 \mathbf{1} & 0 & \cdots & 0 \\ \vdots & D_2 R_2 \mathbf{1} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & D_N R_N \mathbf{1} \end{bmatrix} \tag{14}$$

The results of Figures 3 and Table 1 were obtained using this method.

For problems of the type $-\mathrm{div}(\alpha \nabla u) = f$ with smooth coefficients α, this coarse space gives good results. But for highly heterogeneous coefficients, there is still a plateau in the convergence of the solver. The results in Figure 5 correspond to a domain which has two layers with high values of α. The computational domain has a stripwise decomposition into 64 subdomains. Two Schwarz methods are tested with either no coarse space correction, a Nicolaides coarse space or a spectral coarse space defined in Section 2.2 so a total of six curves. The curves with very long plateaus are one level Schwarz methods. The curves Z_{Nico} (pink curves) correspond to two level Schwarz methods with a Nicolaides coarse space, Equation (13). The plateau in the convergence is not as large but still exists. With the spectral coarse space of the next section, we automatically select two modes per subdomain and get the convergence curves Z_{D2N} (black curves).

2.2 Spectral Coarse Space for Highly Heterogeneous Problems

We now propose a construction of the coarse space that will be suitable for parallel implementation and efficient for accelerating the convergence for problems with highly heterogeneous coefficients of the type:

$$\begin{aligned} -\mathrm{div}(\alpha \nabla u) &= f \quad \text{in } \Omega \\ B(u) &= 0 \quad \text{on } \partial\Omega \end{aligned} \tag{15}$$

with α a positive function. We still choose Z such that it has the form:

$$Z = \begin{bmatrix} W^1 & 0 & \cdots & 0 \\ \vdots & W^2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & W^N \end{bmatrix} \tag{16}$$

where N is the number of overlapping subdomains. But W^i is now a rectangular matrix whose columns are based on the harmonic extensions of the eigenvectors corresponding to small eigenvalues of the Dirichlet-to-Neumann (DtN) map in each subdomain Ω_i. Remark that the sparsity of the coarse operator $E = Z^T A Z$ is a result of the sparsity of Z. The nonzero components of E correspond to adjacent subdomains.

More precisely, let us consider first at the continuous level the Dirichlet-to-Neumann map DtN_{Ω_i} Let u: $\Gamma_i \mapsto \mathbb{R}$, ($\Gamma_i := \partial\Omega_i / \partial\Omega$):

$$\mathrm{DtN}_{\Omega_i}(u) = \alpha \frac{\partial v}{\partial n_i}\bigg|_{\Gamma_i}$$

where v satisfies:

$$\begin{cases} \mathscr{L}(v) := -\mathrm{div}(\alpha \nabla) v = 0, & \text{in } \Omega_i \\ v = u, & \text{on } \Gamma_i \end{cases} \tag{17}$$

and Γ_i is the interface boundary. If the subdomain is not a floating one (*i.e.* $\partial\Omega_i \cap \partial\Omega \neq \phi$), we use on the part of the global boundary, the boundary condition from the original problem $B(u) = 0$. To construct the coarse grid subspace, we use the low frequency modes associated with the DtN operator:

$$\mathrm{DtN}_{\Omega_i}(u) = \lambda \alpha u \tag{18}$$

with:

$$\lambda < 1/diam(\Omega_i) \tag{19}$$

where $diam(\Omega_i)$ is the diameter of subdomain Ω_i. The rationale for this choice is that the condition number

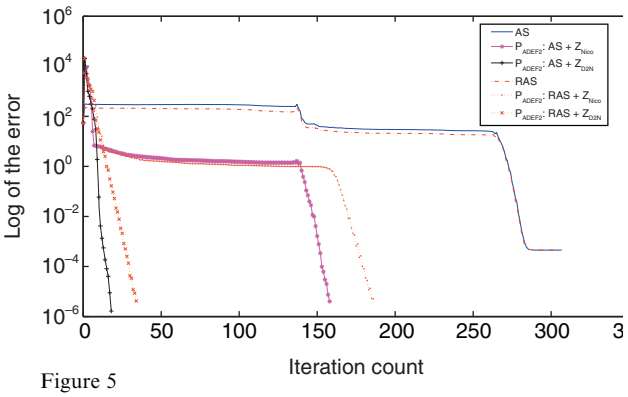

Figure 5

Convergence curves for a domain with two high permeability layers: long plateaus for one level methods, shorter plateaus for Nicolaides coarse spaces and no plateau for DtN coarse space.

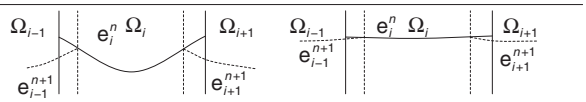

Figure 6

Fast or slow convergence of the Schwarz algorithm.

estimate of Theorem 2.2 is then similar to the one of Theorem 2.1 for the Poisson problem. Note the term α in the generalized eigenvalue problem (18).

We first motivate our choice of a coarse space based on DtN map. We write the original Schwarz method at the continuous level, where the domain Ω is decomposed in a one-way partitioning, Figure 6.

The error e_i^n between the current iterate at step n of the algorithm and the solution $u_{|\Omega_i}$ ($e_i^n := u_i^n - u_{|\Omega_i}$) in subdomain Ω_i at step n of the algorithm satisfies:

$$\begin{aligned} \mathscr{L}(e_i^{n+1}) &= 0 \quad \text{in } \Omega_i \\ e_i^{n+1} &= e_j^n \quad \text{on } \bar{\Omega}_i \cap \partial\Omega_j \end{aligned}$$

On the 1D example sketched in Figure 6, we see that the rate of convergence of the algorithm is related to the decay of the harmonic functions e_i^n in the vicinity of $\partial\Omega_i$ *via* the subdomain boundary condition. Indeed, a small value for this boundary condition leads to a smaller error in the entire subdomain thanks to the maximum principle.

Moreover a fast decay for this value corresponds to a large eigenvalue of the DtN map whereas a slow decay corresponds to small eigenvalues of this map because the DtN operator is related to the normal derivative at the interface and the overlap is thin. Thus the small eigenvalues of the DtN map are responsible for the slow convergence of the algorithm and it is natural to incorporate them in the coarse grid space.

We now explain why we only keep eigenvectors with eigenvalues smaller than $1/diam(\Omega_i)$ in the coarse space. We start with the constant coefficient case $\alpha = 1$. In this case, the smallest eigenvalue of the DtN map is zero and it corresponds to the constant function 1. For a shape regular subdomain, the first positive eigenvalue is of order $1/diam(\Omega_i)$ [12]. Keeping only the constant function 1 in the coarse space leads to good numerical convergence, see Figure 3. In the case of high contrasts in the coefficient α, the smallest eigenvalue of the DtN map is still zero. But due to the variation of the coefficients, we may possibly have positive eigenvalues smaller than $1/diam(\Omega_i)$. In order to have a convergence behavior similar to the one of the constant coefficient case, it is natural to keep all eigenvectors with eigenvalues smaller than $1/diam(\Omega_i)$.

To obtain the discrete form of the DtN map, we consider the variational form of (17). Let's define the bilinear form $a_i : H^1(\Omega_i) \times H^1(\Omega_i) \to \mathbb{R}$:

$$a_i(w, v) := \int_{\Omega_i} \alpha \nabla w \cdot \nabla v$$

With a finite element basis $\{\phi_k\}$, the coefficient matrix of a Neumann boundary value problem in domain Ω_i is:

$$A_{kl}^{(i)} = \int_{\Omega_i} \alpha \nabla \phi_k \cdot \nabla \phi_l$$

Let I (resp. Γ_i) be the set of indices corresponding to the interior (resp. boundary) degrees of freedom and $n_{\Gamma_i} := (\Gamma_i)$ the number of interface degrees of freedom. Note that for the whole domain Ω, the coefficient matrix is given by:

$$A_{kl} = \int_{\Omega} \alpha \nabla \phi_k \cdot \nabla \phi_l$$

With block notations, we have:

$$A_{II}^{(i)} = A_{II},\ A_{\Gamma_i I}^{(i)} = A_{\Gamma_i I} \text{ and } A_{I\Gamma_i}^{(i)} = A_{I\Gamma_i}$$

But the matrix $A_{\Gamma_i\Gamma_i}^{(i)}$ refers to the matrix prior to assembly with the neighboring subdomains and thus cannot be simply extracted from the coefficient matrix A. In problem (17), we use Dirichlet boundary conditions. Let $U \in \mathbb{R}^{n_{\Gamma_i}}$ and $u := \sum_{R \in \Gamma_i} U_R\ \phi_R$. Let $v := \sum_{R \in I} V_R\ \phi_R + \sum_{l \in \Gamma_i} V_l\ \phi_l$ be the finite element approximation of the solution of (17). Let $V_I = (V_k)_{k \in I}$, we have with obvious notations:

$$A_{II} V_I + A_{I\Gamma_i} U = 0 \tag{20}$$

A variational definition of the flux reads:

$$\int_{\Gamma_i} \alpha \frac{\partial v}{\partial n} \phi_k = \int_{\Omega_i} \alpha \nabla v \cdot \nabla \phi_k$$

for all ϕ_k, $k \in \Gamma_i$. So the variational formulation of the eigenvalue problem (18) reads:

$$\int_{\Omega_i} \alpha \nabla v \cdot \nabla \phi_k = \lambda \int_{\Gamma_i} tr(\alpha) v \phi_k \tag{21}$$

for all ϕ_k, $k \in \Gamma_i$ and where $tr(\alpha)$ is the restriction of α_{Ω_i} to Γ_i. Let M_{α,Γ_i} be the weighted mass matrix:

$$\left(M_{\alpha,\Gamma_i}\right)_{kl} := \int_{\Gamma_i} tr(\alpha) \phi_k \phi_l, \quad \forall k, l \in \Gamma_i$$

The compact form of Equation (21) is:

$$A^{(i)}_{\Gamma_i\Gamma_i} U + A_{\Gamma_i I} V_I = \lambda M_{\alpha,\Gamma_i} U$$

With (20), the discrete form of (18) is a generalized eigenvalue problem:

$$\left(A^{(i)}_{\Gamma_i\Gamma_i} - A_{\Gamma_i I} A_{II}^{-1} A_{I\Gamma_i}\right) U = \lambda M_{\alpha,\Gamma_i} U \tag{22}$$

Let (U_λ, λ) be an eigenpair, we need its harmonic extension to the degrees of freedom of domain Ω_i, that is the vector:

$$\begin{bmatrix} -A_{II}^{-1} A_{I\Gamma_i} U_\lambda \\ U_\lambda \end{bmatrix}$$

Actually, there is more practical way to "directly" compute these eigenpairs. For subdomain Ω_i, let:

$$v := \begin{bmatrix} V_I \\ V_{\Gamma_i} \end{bmatrix}, \quad A^{(i)} := \begin{bmatrix} A_{II} & A_{I\Gamma_i} \\ A_{\Gamma_i I} & A^{(i)}_{\Gamma_i\Gamma_i} \end{bmatrix}$$

we compute the lowest eigenvalues of the sparse generalized eigenvalue problem:

$$A^{(i)} v_{dtn} = \lambda \begin{bmatrix} 0 & 0 \\ 0 & M_{\alpha,\Gamma_i} \end{bmatrix} v_{dtn} \tag{23}$$

This can be done using standard linear algebra library such as ARPACK.

The step by step procedure on how to construct the rectangular matrices W^i in the coarse space matrix Z, see (16), is summed up in Algorithm 1.

Algorithm 1. Construction of the spectral coarse space
In parallel for all subdomains $1 \le i \le N$,

1. Compute eigenpairs of (23) $\left(V^i_1, \lambda^i_1\right), \left(V^i_2, \lambda^i_2\right), \ldots, \left(V^i_{m_i}, \lambda^i_{m_i}\right)$ such that:

$$\lambda^i_1 \le \cdots \le \lambda^i_{m_i} < \frac{1}{diam(\Omega_i)} \le \lambda^i_{m_{i+1}} \le \cdots$$

2. Let Z de defined as in (16) with for each $1 \le i \le N$, W^i the rectangular matrix with m_i columns defined by:

$$W^i = \left[D_i V^i_1 | \cdots | D_i V^i_{m_i}\right]$$

3. Note $R_0 := Z^T$ and compute the coarse matrix E:

$$E := R_0 A R_0^T$$

4. The two-level preconditioner is given by Equation (11):

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

We call this procedure the Z_{D2N} method. We also use Z_{D2N} to denote the coarse space constructed by this method. Its construction is fully parallel. Similarly we call Z_{Nico} the method of Nicolaides or the corresponding coarse space. Let us remark that when the subdomain does not touch the boundary of Ω, the lowest eigenvalue of the DtN map is zero and the corresponding eigenvector is a constant vector. Thus, Z_{Nico} and Z_{D2N} coincide. As we shall see in the next section, when a subdomain has several jumps of the coefficient, taking Z_{Nico} is not efficient and it is necessary to take Z_{D2N} with more than one mode per subdomain.

This construction has been analyzed in [13]. We first recall a classical result. Let Z be a "Nicolaides type" coarse space:

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

We have, see [1]:

Theorem 2.1. *Let $M_{(ASM,2)}$ be the two-level additive Schwarz method with the "Nicolaides" coarse space, we have for $\alpha = 1$ the following condition number estimate:*

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

where δ is the size of the overlap between the subdomains and H the subdomain size and C does not depend on the number of subdomains.

But, for α discontinuous, C would depend on the jumps of α.

Let Z be the coarse space built *via* Algorithm 1, we prove under technical assumptions on α:

Theorem 2.2. *Under the monotonicity of α in the overlapping regions, we have the following condition number estimate:*

$$\kappa\left(M^{-1}_{ASM,2} A\right) \le C\left(1 + \max_{1 \le i \le N} \frac{1}{\delta_i \lambda^i_{m_i+1}}\right)$$

where δ_i is the size of the overlap of domain Ω_i and C is independent of the jumps of α and of the number of subdomains.

Note that if $\alpha = 1$ and we take only one mode per subdomain ($m_i = 1$), we have for a regular interface

$\lambda_2^i \simeq 1/H_i$ [12] and we recover the "classical" estimate. Now in the general case, if the number of modes associated to subdomain Ω_i m_i is chosen so that, $\lambda_{m_i+1}^i \geq 1/H_i$, the convergence rate will be analogous to the constant coefficient case.

2.3 Comparison with a Volumic Spectral Coarse Space

The DtN spectral coarse space makes use of eigenvectors of the local Dirichlet to Neumann maps. There is thus a clear relationship with recent works by Galvis and Efendiev [14-17] where the coarse space is based on eigenvalues of the "volumic" operator:

$$-div(\alpha \nabla u_i) = \lambda \alpha u_i \text{ in } \Omega_i \tag{24}$$

The drawback of their approach is that the coarse space is too large. This is easy to see in 1D. In Figure 7, we represent the function α in a subdomain Ω_i. We have many discontinuities inside the domain. But, whatever the number of discontinuities is, our DtN map is a two by two matrix. The number of eigenvectors of the DtN map is two. Thus, the coarse space is made of two vectors per subdomain at most. But, the size of a volumic spectral coarse space is equal to the number of high heterogeneities islands. This phenomena also holds in the 2D case. In Figure 8, we show permeability field with high heterogeneities islands. For this case, only 4 eigenvalues of the DtN map are smaller than 1.3×10^{-4} whereas the other ones are larger than 0.9. Whereas twenty volumic eigenvalues of Equation (24) are smaller than 3.8×10^{-3} and the others are larger than 150. Numerical tests show that in this case only four eigenvalues are enough for having an efficient coarse space. In the papers by Galvis and Efendiev [17], they noted this fact and they have a complex procedure to get rid of the useless eigenvectors. In our case, the method adapts automatically to the permeability field. In Figure 9, we show typical DtN and volumic eigenvectors.

2.4 First Numerical Tests

We solve the model problem (15) on the domain $\Omega = [0, 1]^2$ using standard continuous, piecewise linear (P_1) finite elements. The diffusion α is a function of $\mathbf{x}$. The boundary condition is $u = 0$ on the left side boundary and $\frac{\partial u}{\partial n} = 0$ on the remainder. The corresponding discretizations and data structures were obtained by using the software FreeFem++ [18] in connection with the METIS partitioner [19]. We will test the Standard Additive

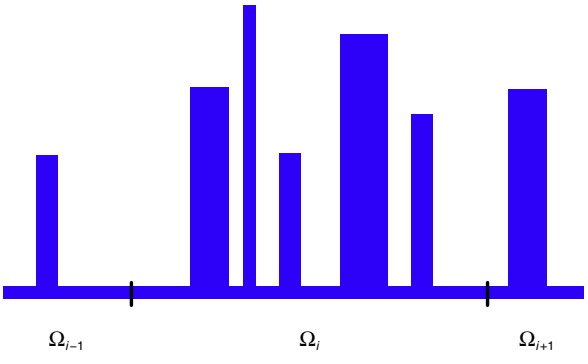

Figure 7

1D example with many high heterogeneities islands.

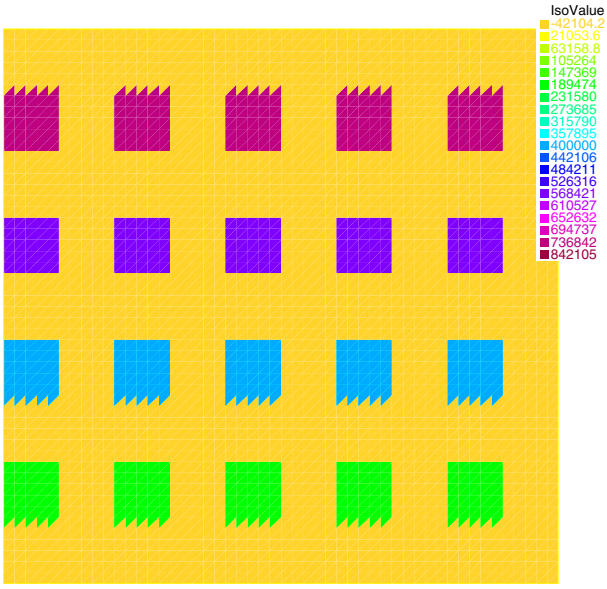

Figure 8

2D example with many high heterogeneities islands.

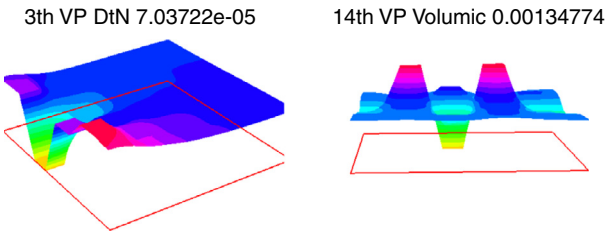

Figure 9

Eigenvectors for a) DtN map and b) the volumic operator (Free Fem++ plots).

TABLE 2
Iteration counts *vs* jumps in the coefficients

Jumps in coeff	1	10	10^2	10^3	10^4	10^5	10^6
Iteration counts	15	24	10	10	10	11	11

Figure 10
Subdomain partitioning into 16 subdomains using METIS.

Schwarz (SAS) and the RAS preconditioners with and without coarse space, in particular comparing the new coarse space based on harmonic extensions of eigenvectors of the local DtN operators with the standard coarse space that is the piecewise constant space of Nicolaides [11]. In the tables and figures, +Nico means the use of the Nicolaides coarse space (14) and +DtN the use of the spectral coarse space defined in Algorithm 1. We test the method on (fairly irregular) overlapping partitions into N subdomains. These overlapping partitions are built by adding layers to non-overlapping ones obtained, *e.g.*, *via* graph partitioner Metis (*Fig. 10*).

In Table 2, we test robustness w.r.t. the heterogeneities. The domain Ω contains layers with jumps in the coefficients ranging from 1 to 10^6. We have 32 subdomains. The iteration counts depend weakly on the size of the jump in the coefficients. In Figure 11, we show the permeability field, domain decomposition (regular or METIS) into 16 subdomains and the solution corresponding to convergence curves of Figures 12 and 13. In Table 3, we show how many eigenvalues were selected in the coarse space. In Table 4, we vary the domain decomposition for the same permeability field.

We now present a selection of difficult test cases in a more systematic way, with so called inclusions and channels.

We solve two test cases with known difficulties. The diffusion coefficient α is highly heterogeneous and takes values between 1 and approximately 2×10^6 and contains both high-permeability inclusions and channels. First of all, we will analyze the performance of the method by increasing the number of channels and then by increasing the number of inclusions.

We use a uniform triangulation with 160×160 nodes and a partition into 16 (irregular) subdomains (*Fig. 10*). Each subdomain is extended by one layer, leading to an overlap of 2 layers, such that $\delta_j = 2h$ for all $j = 1, \ldots, N$. We use the ASM preconditioner within Conjugate Gradients (CG) and the RAS preconditioner within GMRES, and in each case, we stop the iteration process, when the relative residual is smaller than 10^{-6}.

We start with only inclusions and add the channels one by one as shown in Figure 14 (Test Problem 1). When there are no channels, α varies between 1 and 10^6, as indicated by the colors in Figure 14. With all three channels present, α varies between 1 and 2.8×10^6. The corresponding convergence results are given in Table 5. Our algorithm performs significantly better. The piecewise constant coarse space has virtually no effect on the performance of either ASM or RAS, leading to iteration numbers that differ little from the results without any coarse grid in all four cases. Our new coarse space, on the other hand, is fully robust to the coefficient variation and to the addition of channels, and it leads to a gain of at least a factor 8 compared to the one-level method in all cases. The situation is similar, if we use deflation-based coarse grid correction [10] with the same coarse spaces (*Tab. 6*). However, the absolute numbers of iterations are reduced almost by a factor 2 in this case. Our theory applies equally to this case [20], but we will not include any further numerical results with deflation-based coarse grid correction.

Table 7 gives some information on the size of the coarse space that we build with our automatic selection strategy: for each number of channels we give $\min_j m_j$ and $\max_j m_j$, as well as the global coarse space size $n_H = \sum_j m_j$ and the average number of modes included per subdomain n_H/N. For comparison, we also include information on the total number n_{Γ_j} of eigenmodes of the discrete DtN operator on each subdomain. We note that adding channels does not have a big influence on the

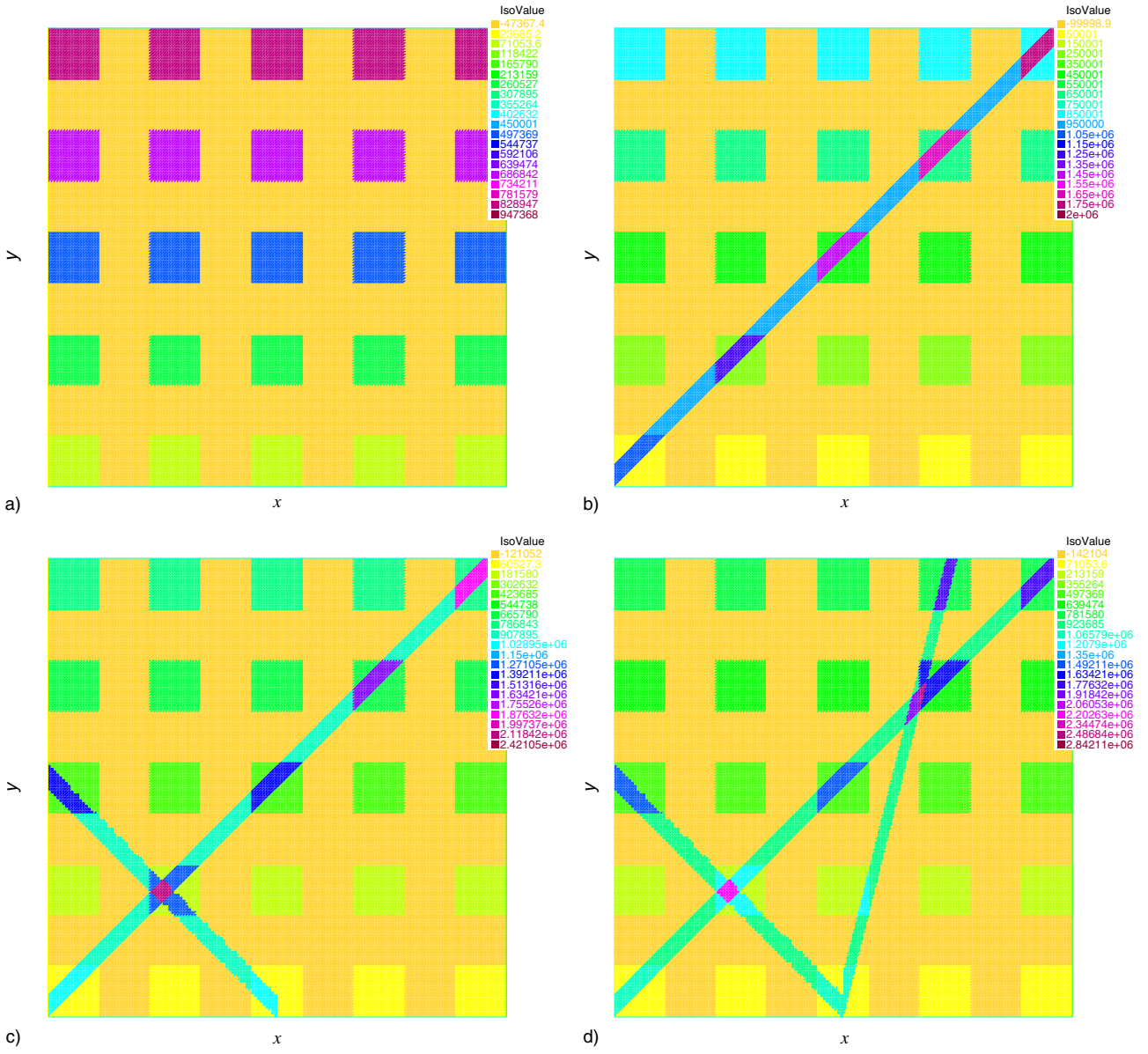

Figure 11

Channels and inclusions: $1 \leq \alpha \leq 1.5 \times 10^6$. a) permeability field, b) the solution, c) regular partition and permeability field, d) Metis partition and permeability field.

size of the coarse space; we only need three additional eigenvectors in the case of three channels compared to the case of no channels.

Then, using the same domain and the same partition we successively add inclusions without any channels present as shown in Figure 15 (Test Problem 2). The results are in Table 8. Again, the piecewise constant coarse space is not working at all for this test problem. The DtN-based coarse space is almost completely robust to an increase in the number of inclusions and requires again significantly less iterations than the one-level method in all cases. Note that the subdomain partition (*Fig. 10*) is not aligned with the inclusions at all (*Fig. 15*). In Table 9, we see that also in this test problem, the coarse space size grows only very slowly with the number of inclusions and even in the hardest case n_H is

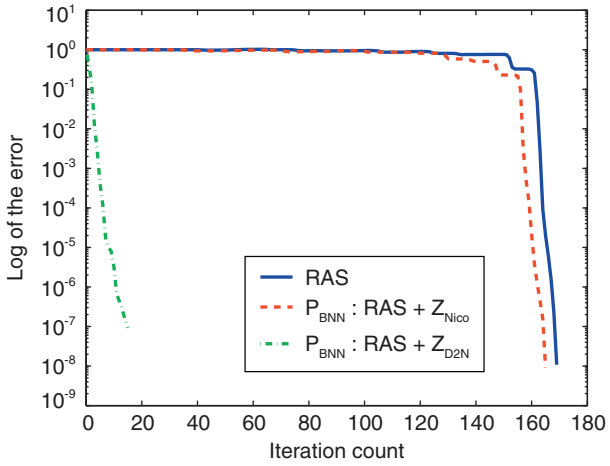

Figure 12

RAS convergence for channels and inclusions – Regular partitioning.

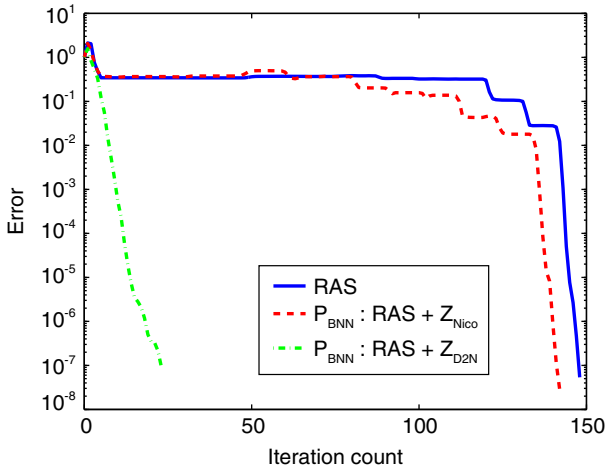

Figure 13

RAS convergence for channels and inclusions – Regular decomposition – Metis partitioning.

TABLE 3

Number of small eigenvalues (nsmeig(i)) satisfying criterion (19) for subdomain i – Metis 4 by 4 decomposition

Subdomain i	*nsmeig*(i)	Total number of eigenvalues (i)
1	3	155
2	1	109
3	5	175
10	4	174
11	2	71
12	2	128
13	3	166
14	3	127
15	3	188
16	3	106

TABLE 4

Convergence results for the test case of Figure 10

	ASM	+ Nico	+ DtN	RAS	+ Nico	+ DtN
2 × 2	103	110	22	70	70	14
2 × 2 Metis	76	76	22	57	57	18
4 × 4	603	722	26	169	165	15
4 × 4 Metis	483	425	36	148	142	23
8 × 8	461	141	34	205	95	21
8 × 8 Metis	600	542	31	240	196	19

only 53 (cf. the dimension n of $V_{h,0}$, and thus of A is 25 600).

2.4.1 Practical Optimality of the Spectral Coarse Space

The last series of tests, in Table 10, aims to prove that the number m_j of eigenvectors per subdomain chosen by our automatic algorithm is indeed optimal in some sense. For Test Problem 1 with one channel (*Fig. 14*), we first reduce the number of coarse basis functions per subdomain by one, this has a huge influence on the iteration count. Then, we add one basis function per subdomain and notice that this has much less effect. This suggests that the selection process we have designed is indeed the best compromise between enriching the coarse grid and solving a reasonably sized coarse problem.

3 ADAPTIVE COARSE SPACE ON HPC PLATFORMS

Results in this section are based on a related method to the DtN coarse space method namely the Geneo method. The principle of this coarse space construction is similar in that the coarse space is built after solving local eigenvalues problems. It suffices to change the right

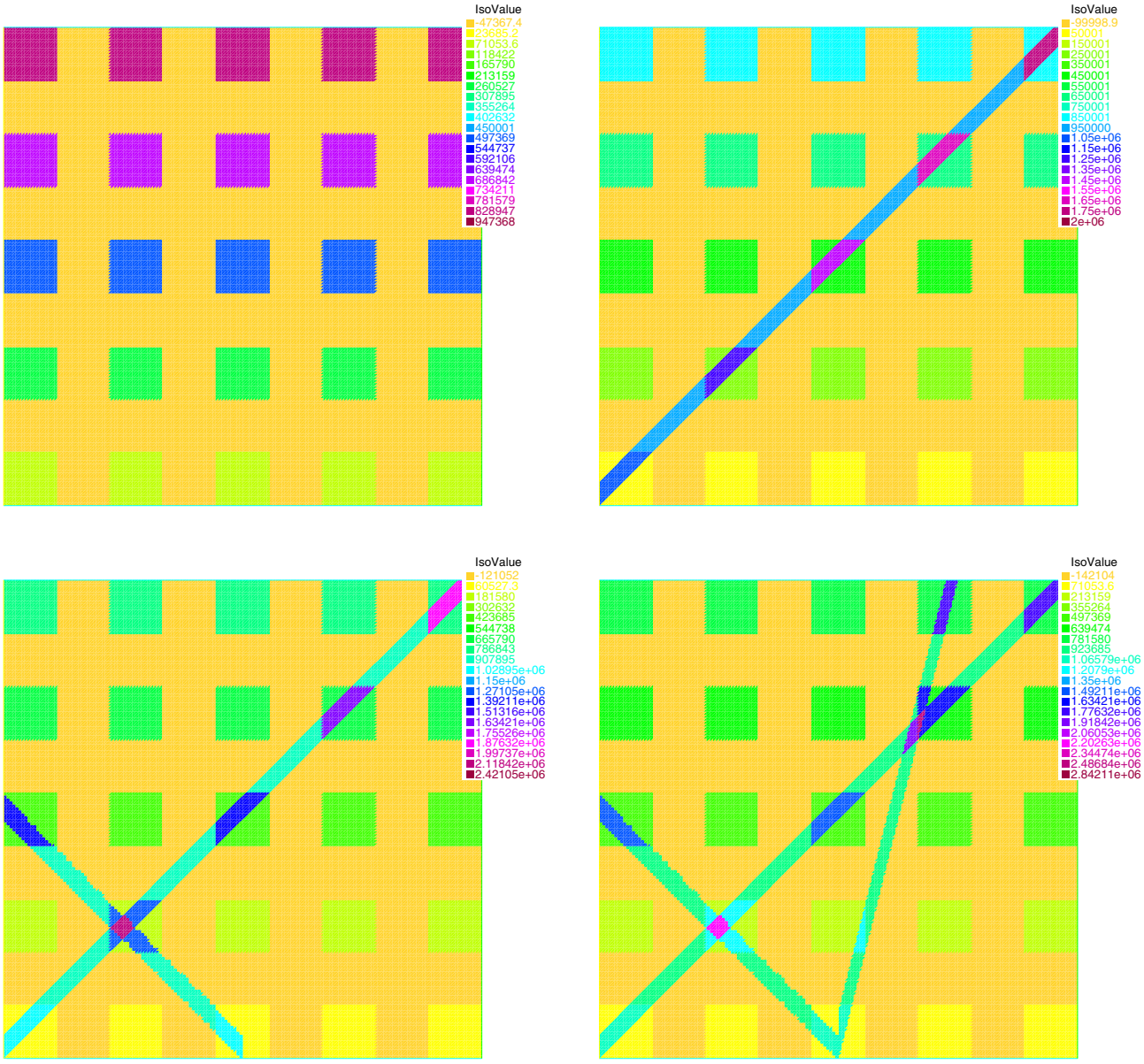

Figure 14

Test Problem 1: successively adding channels.

hand side in the generalized eigenvalue problem (23). The new eigenvalue problem is of the form:

$$A^{(i)}v_{dtn} = \lambda D_i R_i A R_i^T D_i v_{dtn} \qquad (25)$$

see [21] for more details. The Geneo coarse space is in practice quite close to the DtN coarse space. Its main advantage is to work not only for scalar PDE but also for systems of PDE as the elasticity system for instance. When applied to scalar PDE, DtN and Geneo coarse spaces are almost identical and give very similar results. As a result, in order to have a general purpose code, we focused in HPC developments and tests on the Geneo method. Results in this section were obtained on Curie, a Tier-0 system for PRACE2 (Partnership for Advanced Computing in Europe) composed of 5 040 nodes made of 2 eight-core Intel Sandy Bridge processors clocked at 2.7 GHz. The interconnect is an InfiniBand QDR full

TABLE 5
Number of iterations for Test Problem 1 (additive coarse grid correction)

	ASM	+ Nico.	+ DtN	RAS	+ Nico.	+ DtN
0 ch.	529	1 000	57	243	245	41
1 ch.	619	520	64	227	228	46
2 ch.	> 1 000	516	68	226	226	47
3 ch.	585	697	76	212	213	44

TABLE 6
Number of iterations for Test Problem 1 (deflation-based coarse grid correction)

	ASM	+ Nico.	+ DtN	RAS	+ Nico.	+ DtN
0 ch.	529	656	39	243	231	25
1 ch.	619	538	41	227	215	28
2 ch.	> 1 000	808	47	226	211	27
3 ch.	585	641	47	212	199	28

TABLE 7
Size of the coarse space for Test Problem 1 with various number of channels

Over 16 subdomains	# eigenvalues on Γ_j	# Local coarse space modes			
		0 ch.	1 ch.	2 ch.	3 ch.
Minimum	70	1	1	1	1
Maximum	191	4	4	4	4
Average	138.8	2.75	2.88	2.94	2.94
Sum	2 220	44	46	47	47

fat tree. We want here to assess the capability of our framework to scale:

- strongly: for a given "global" mesh, the number of subdomains increases while "local" mesh sizes are kept constant (*i.e.* local problems get smaller and smaller);
- weakly: for a given "global" mesh, the number of subdomains increases while "local" mesh sizes are refined (*i.e.* local problems have a constant size).

We don't time the generation of the mesh and partition of unity. Assembly and factorization of the local stiffness matrices, resolution of the generalized eigenvalue problems, construction of the coarse operator and time elapsed for the convergence of the Krylov method are the important procedures here. The Krylov method used is the GMRES, it is stopped when the relative residual error is inferior to $\varepsilon = 10^{-6}$ in 2D, and 10^{-8} in 3D. All the following results where obtained using a LDL^T factorization of the local solvers A_i^δ and the coarse operator E using MUMPS (with a MPI communicator set to respectively MPI_COMM_SELF or the communicator created by our library binding master processes).

First, the system of linear elasticity with highly heterogeneous elastic moduli is solved with a minimal geometric overlap of one mesh element. Its variational formulation reads:

$$\int_\Omega \lambda \nabla \cdot u \nabla \cdot v + 2\mu\varepsilon(u)^T\varepsilon(v) + \int_\Omega f \cdot v + \int_{\partial\Omega} g \cdot v \quad (26)$$

where:

- λ and μ are the Lamé parameters such that $\mu = \frac{E}{2(1+\nu)}$ and $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ (E being Young's modulus and ν Poisson's ratio). They are chosen to vary between two set of values, $(E_1, \nu_1) = (2 \times 10^{11}, 0.25)$, and $(E_2, \nu_2) = (10^8, 0.4)$;
- ε is the linearized strain tensor and f the volumetric forces (here, we just consider gravity).

Because of the overlap and the duplication of unkowns, increasing the number of subdomains means that the number of unknowns increases also slightly, even though the number of mesh elements (triangles or tetrahedra in the case of FreeFem++) is the same. In 2D, we use piecewise cubic basis functions on an unstructured "global" mesh made of 110 million elements, and in 3D, piecewise quadratic basis functions on an unstructured "global" mesh made of 20 million elements. This yields a symmetric system of roughly 1 billion unkowns in 2D and 80 million unknowns in 3D. The geometry is a simple $[0; 1]^d \times [0; 10]$ beam ($d = 1$ or 2) partitioned with Metis.

Solving the 2D problem initially on 1 024 processes takes 227 seconds, on 8 192 processes, this time is reduced to 31 seconds (quasioptimal speedup) (*Fig. 16*). With that many subdomains, the coarse operator E is of size 121 935 × 121 935. It is assembled and factorized in 7 seconds by 12 master processes. For the 3D problem, the wall-clock time is initially 373 seconds. At peak performance, near 6 144 processes, the time is reduced to 35 seconds (superoptimal speedup). Then, the coarse operator is of size 92 160 × 92 160 and is assembled and factorized by 16 master processes in 11 seconds.

Moving on to the weak scaling propreties of our framework, (*Fig. 17*) the problem we now solve is a scalar equation of diffusivity with highly heterogeneous

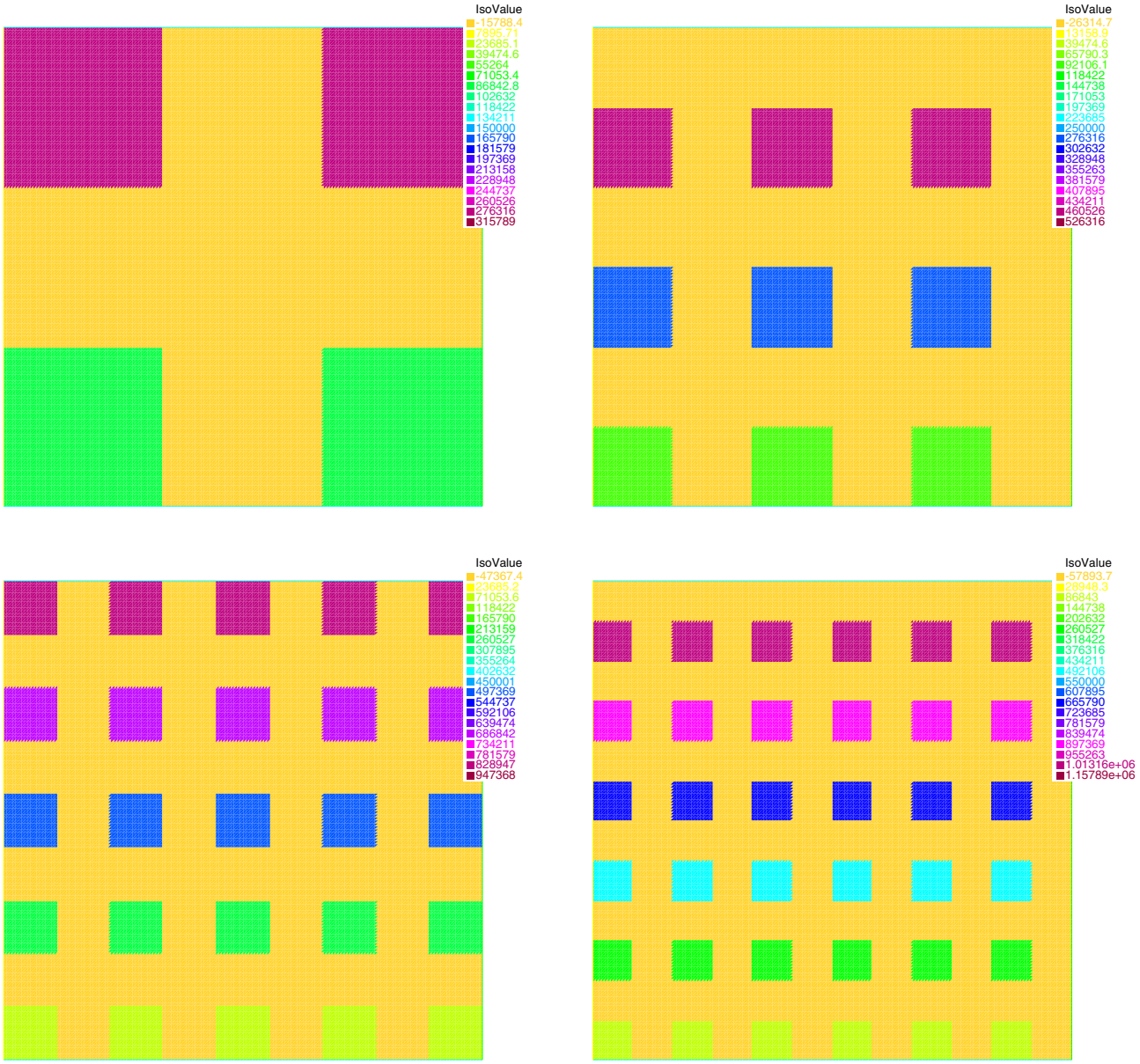

Figure 15

Test Problem 2: successively adding inclusions.

coefficients (varying from 1 to 10^5) on $[0;1]^d$ ($d = 2$ or 3). Its variational formulation reads:

$$\int_\Omega \alpha \nabla u \cdot \nabla v + \int_\Omega f \cdot v \tag{27}$$

The targeted number of unkowns per subdomains is kept constant at approximately 800 thousands in 2D, and 120 thousands in 3D (once again with $\mathbb{P}_3$ and $\mathbb{P}_2$ finite elements respectively).

In 2D, the initial extended system (with the duplication of unkowns) is made of 800 million unkowns and is solved in 141 seconds. Scaling up to 12 288 processes yields a system of 10 billion unkowns solved in 172 seconds, hence an efficiency of $141/172 \approx 82\%$. In 3D, the initial system is made of 130 million unkowns and is solved in 127 seconds. Scaling up to 8 192 processes yields a system of 1 billion unkowns solved in 152 seconds, hence an efficiency of $127/152 \approx 83\%$.

TABLE 8
Number of iterations for Test Problem 2 (additive coarse grid correction) *vs* number of inclusions

# incl.	ASM	+ Nico	+ DtN	RAS	+ Nico	+ DtN
2 × 2	108	80	51	100	81	41
3 × 3	194	342	58	154	153	46
5 × 5	529	no cv.	57	243	245	41
6 × 6	835	823	71	266	267	51

TABLE 9
Size of the coarse space for Test Problem 2

Over $N = 16$ subdomains	# eigen values on Γ_j	# Local coarse space modes			
		0 ch.	1 ch.	2 ch.	3 ch.
Minimum	70	1	1	1	1
Maximum	191	3	3	4	5
Average	138.8	1.6	2.1	2.8	3.3
Sum	2 220	26	33	44	53

TABLE 10
Iteration numbers when reducing or increasing the number m_j of coarse basis functions per subdomain given by the automatic selection strategy

	ASM	RAS
No coarse space	619	227
Piecewise constant coarse space	520	228
DtN with $\max\{m_j - 1,1\}$ functions	446	177
DtN with m_j functions	**64**	**46**
DtN with $m_j + 1$ functions	37	32

4 CONNECTIONS WITH MULTISCALE METHODS

Multiscale methods are an active field of research, for finite element methods see [22] and for multiscale finite volume methods see for example [23]. In Section 4, we compare them with our two-level spectral coarse space. In Section 4.1, we first recall basic facts on multiscale discretizations and their difficulties with arbitrary channelized flows, Section 4.1.1.1. Although the goals of multiscale methods and DD methods are different, they have many related features that we compare in Section 4.2. In particular, both methods build coarse basis functions. The superiority of the spectral coarse space comes the fact that the "number" and the "shape" of basis functions adapts automatically to the heterogeneities of the medium even for channelized media. This is not always the case for multiscale methods.

4.1 Presentation of Multiscale Methods

Consider a problem set on a fine grid (*Fig. 18*):

$$\mathcal{L}_h(u_h) = f_h \text{ in } \Omega_h \tag{28}$$

that is too large to be solved. We approximate u_h *via* a coarse problem set on a coarse mesh Ω_H. Defining a multiscale methods involve three steps:
- pre-computation of a multiscale basis functions;
- global formulation at the coarse level;
- reconstruction of a fine scale solution.

There are of course many variants to deal with these topics and we don't try to give a complete review on the subject. We present here basic materials in order to compare multiscale methods with our DtN coarse space. In particular, we shall see that the DtN approach is more general and systematic.

4.1.1 Multiscale Basis Functions

The preferred and most common technique is to use oversampling, [22]. For simplicity, we start with the original non oversampling approach.

We consider a structured two-dimensional grid. A coarse element is typically denoted by K Let (x_i, y_i) be a coarse grid vertex. We recall the construction of the corresponding coarse basis function $\varphi_{H,i,j}$. For both Multiscale Finite Element Method (MsFEM) and Multiscale Finite Volume (MsFV), a standard choice is to solve the fine scale equation on the four neighboring coarse elements $K_{i\pm 1/2,j\pm 1/2}$, Figure 18:

$$\begin{aligned} \mathcal{L}_h\left(\phi_{i\pm 1/2,j\pm 1/2}\right) &= 0 \quad \text{in} \quad K_{i\pm 1/2,j\pm 1/2} \\ \phi_{i\pm 1/2,j\pm 1/2} &= g_{i\pm 1/2,j\pm 1/2} \quad \text{on} \quad \partial K_{i\pm 1/2,j\pm 1/2} \end{aligned} \tag{29}$$

where $g_{i\pm 1/2,j\pm 1/2}$ is a piecewise affine function such that $g_{i\pm 1/2,j\pm 1/2}(x_i,y_i) = 1$ and is zero on the three other vertices of $\partial K_{i\pm 1/2,j\pm 1/2}$. Then, function $\phi_{H,i,j}$ is defined by taking restrictions of $\phi_{i\pm 1/2,j\pm 1/2}$ to the coarse elements adjacent to the coarse grid vertex (x_i, y_j):

$$\phi_{H,i,j}(x,y) = \begin{cases} \phi_{i\pm 1/2,j\pm 1/2}(x,y) & \text{if } (x,y) \in K_{i\pm 1/2,j\pm 1/2} \\ 0 & \text{Otherwise} \end{cases} \tag{30}$$

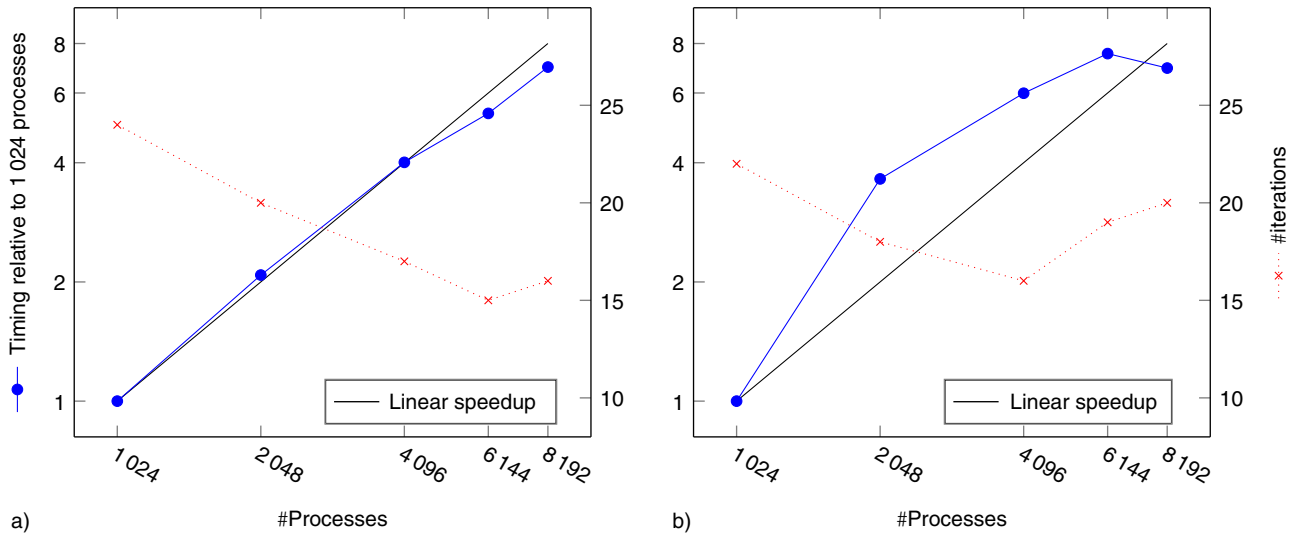

Figure 16

Linear elasticity test cases. a) 2D, b) 3D. Strong scaling.

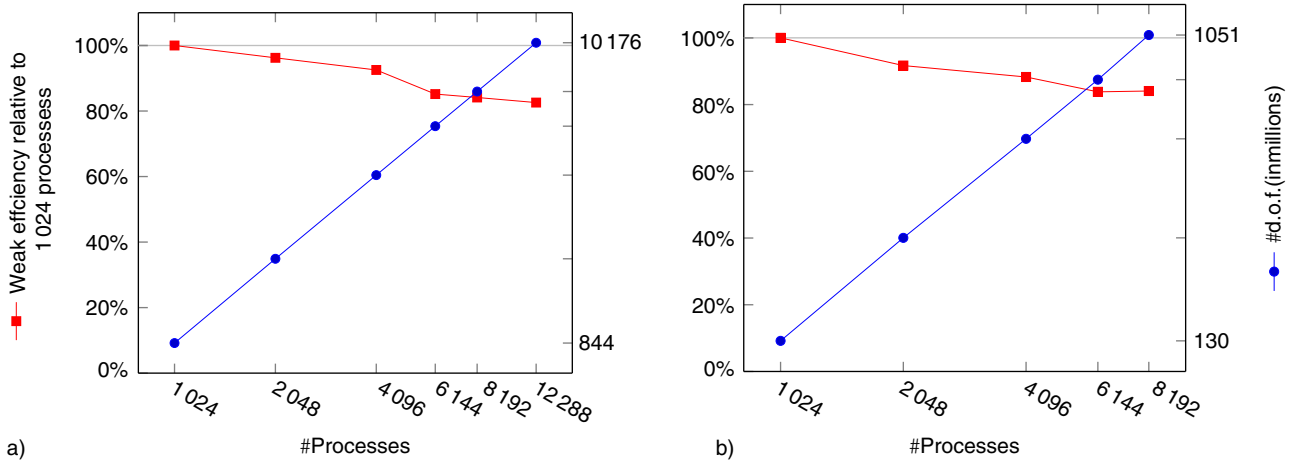

Figure 17

Diffusion equation test cases. a) 2D, b) 3D. Weak scaling.

This construction presents unwanted boundary layers effects. In order to fix this problem, functions $\phi_{i\pm1/2,\,j\pm1/2}$ are computed on a coarse cell $K^{\partial}_{i\pm1/2,j\pm1/2}$ enlarged with a few layers of fine elements, Figure 18. Then a coarse basis function $\phi_{H,i,j}(x, y)$ is computed as a linear combination of the restrictions of functions $\phi_{i\pm1/2,\,j\pm1/2}$ to $K_{i\pm1/2,\,j\pm1/2}$. This leads to a non conformal basis. When the coefficients of the operator $\mathscr{L}_h$ are sufficiently smooth, this basis is adequate. This procedure is called oversampling.

When the coefficients are heterogeneous across these edges (*Fig. 19a*) the basis functions should see the heterogeneities. For this purpose, the piecewise linear Dirichlet boundary conditions are replaced by oscillatory boundary condition obtained by solving a reduced elliptic problem along the boundary of the coarse cell. The Dirichlet data must be in the kernel of the tangential part of the partial differential operator in Equation (29). An algebraic implementation of this construction was proposed in [24] and [25].

Note that for finite volume schemes for problems with high anisotropies, the cell problems (29) can also be modified by replacing Dirichlet Boundary Conditions (BC)

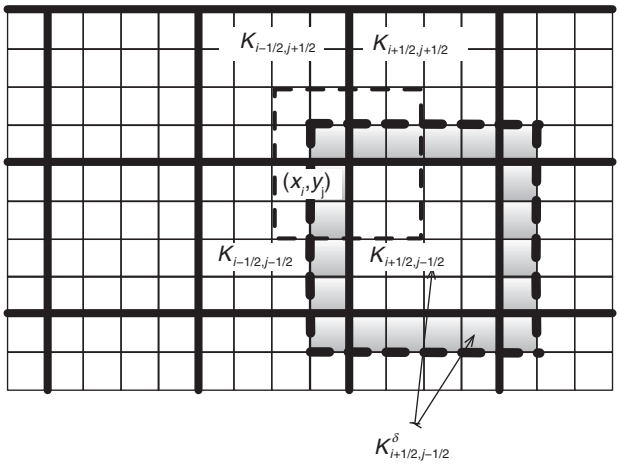

Figure 18

Fine mesh Ω_h, coarse mesh Ω_H and a dual coarse cell around point (x_i, y_j).

by Neumann BC on some parts of $\partial K^{\partial}_{i\pm1/2,j\pm1/2}$, see Section 6.3. of [23].

4.1.1.1 Multiscale Basis Functions for Channelized Permeability Distributions

When the problem has strong heterogeneities, typically, three situations occur as shown in Figure 19:
- isolated heterogeneities;
- one heterogeneous channel;
- several heterogeneous channels;

The first case is well treated by multiscale methods as recalled above. For the second case, it has been noticed that it might not be sufficient: "It has been shown that the accuracy of purely local methods may be low if the permeability field has structures with very long correlation lengths" quoted from [23]. This is the case for instance with channels, Figure 19b,c. In order to fix this problem, iterative constructions of the coarse basis functions have been proposed, [26-28]. Iterations take place between the coarse scale global flow and the fine scale local flow. A coarse space is first built with the oversampling method. It is used to obtain a coarse and then a fine grid solution. Then, the coarse basis functions are corrected by taking the coarse edge values of this solution as Dirichlet boundary conditions in Equation (29). This procedure stops with some convergence criterion. To our knowledge, this technique is not supported by theoretical approximation results. The last case with several heterogeneous channels seems to be a concern even for this approach. Indeed,

Figure 19

a) Isolated heterogeneities, b) one channel, c) two channels.

in this case the good coarse space functions depend on the flow conditions: "The introduction of wells may additionally change global flow significantly and the coarse properties generated from the two generic global flows might lose accuracy in some cases. For such problems, the T can be recomputed, based on the actual well configuration and flow rates, using a local-global procedure analogous to that applied here. The overall issue of robustness with respect to global boundary conditions is complex and will be addressed in detail in a future paper." quoted from [26]. The problem comes from the fact that the "number" of coarse basis functions attached to the cell should be at least equal to the "number" of channels crossing the cell, see [29]. But, in multiscale methods even in the more algebraic ones as [25], the number of degree of freedom per aggregate is prescribed in advance. For a scalar problem, only one coarse basis function is assigned to a coarse grid vertex. It is thus not possible to cover all possible flow configurations.

4.1.2 Coarse Problem

This step consists in approximating the fine scale solution u_h by defining a suitable coarse space problem whose solution, denoted by u_H, belongs to the space spanned by the coarse basis functions $(\phi_{H,i,j})_{i,j}$. We consider first finite element formulation and then finite volume approximations.

For a finite element method, a Galerkin approach is usually used. For all i,j, let us denote by $Z_{i,j}$ the vector of the components of $\phi_{H,i,j}$ on the basis of the fine FEM. We collect all these vectors in a rectangular matrix Z. Let A_h denote the matrix associated to the fine FEM so that the matrix form of the fine FEM reads:

$$A_h U_h = F_h \tag{31}$$

where U_h are the components of the solution u_h on the fine FEM basis. Let us define $A_H := Z_t\, A_h\, Z$ and the coarse problem by:

Find $u_H := \sum_{i,j} U_{H,i,j}\phi_{H,i,j}$ such that:

$$A_H U_H = Z^T F_h$$

This way, the coarse approximation U_H satisfies a variational formulation in the coarse space spanned by the coarse basis functions $\phi_{H,i,j}$.

For finite volume methods, the Galerkin approach can be used as well. But then, conservativity and monotonicity of the initial finite volume scheme are lost. In order to recover them, a dual coarse mesh is introduced, Figure 18. The coarse grid problem consists in finding $u_H := \sum_{i,j} U_{H,i,j}\phi_{H,i,j}$ such that conservativity is satisfied on the boundaries of the dual cells. Typically, a 9-point stencil is thus obtained and for anisotropic problems the monotonicity of the finite volume scheme on the fine mesh is lost on the coarse problem. Then, a modified 7-point stencil is sought that still ensures conservativity, [24].

4.1.3 Fine Scale Solution

This step is actually optional since U_H contains fine scale information *via* the coarse basis functions $\phi_{H,i,j}$. In MsFEM, one can further improve the solution by solving local Dirichlet boundary value problems in each coarse element $K_{i\pm 1/2,\ j\pm 1/2}$

$$\mathcal{L}_h(u_h) = f_h \quad \text{in} \quad K_{i\pm 1/2, j\pm 1/2}$$

and

$$u_h = U_H \quad \text{on} \quad \partial K_{i\pm 1/2, j\pm 1/2}$$

Thus U_H could be used in principle to solve for instance a transport equation at the fine level. But the method to compute U_H is not conservative which is then a big drawback. In multiscale finite volume methods, the reconstruction is based on solving Neumann problems in each coarse cell so that local conservativity is satisfied. As a result, the fine scale solution is not continuous at the edges of the coarse elements.

4.2 Comparison with the DtN Two-Level Schwarz Method

The foremost difference between multiscale methods and two-level domain decomposition methods is the goal itself. In the first method, one wants to "approximate" the solution of the fine scale problem whereas in the second method one wants to "solve" the fine scale Equations (28) or equivalently Equation (31). In this respect, multiscale methods are competitors to homogenization or upscaling methods, [22]. But in contrast to these methods, multiscale methods don't lead to some kind of average PDE models. They are a framework to provide a cheap way *via* a coarse solve to approximate the solution u_h of a (too) large scale system of equations. In this respect, they can be seen as approximate two level solvers and could be used as well as preconditioners for Krylov type methods such as CG, GMRES or BICG-STAB. They are thus naturally comparable to two level DD methods such as the DtN approach described above. This has been noticed by several authors, [24, 25, 30]. There, the multiscale approach is simply a framework to provide an adequate coarse space. Thus, we compare the coarse basis functions constructions and give indications of their relative efficiency as preconditioners.

Moreover, the involved tools have some similarities.

4.2.1 Oversampling and Overlapping

For both methods, the fine mesh is decomposed into aggregates of fine elements. But:

- in MsFEM or MsFV, the aggregates consist of some dozens of elements;
- whereas in DDM, subdomains may be quite large the construction of the coarse problem is essentially parallel.

But:

- in multiscale methods, we have a fine grain parallelism;
- in DDM, we have a coarse grain parallelism.

Oversampling is very reminiscent of overlapping in DDM. In both approaches, coarse basis functions are harmonic functions in overlapping aggregates (subdomains in DDM and extended coarse cells in oversampling multiscale methods). In order to use them in a coarse problem, they have to be cast to functions defined in the whole domain Ω_h. In multiscale methods this was done *via* procedure which is somehow "brutal" since the resulting function is not even continuous on Ω_h. In DD methods, the local coarse space functions are multiplied by a kind of partition of unity (the local matrices $(D_i)_{1 \le i \le N}$, see formula (12)) before the extension by zero in the whole domain Ω_h so that the resulting function is continuous on Ω_h. In [15], the authors use partition of unity functions in MsFEM methods.

4.2.2 Local and Global Effects

Coarse basis functions are used to define a coarse problem and they are harmonic in the aggregates. But:

- in multiscale methods, they mimic finite element basis function: only one such function per aggregate;
- in the spectral coarse space of Section 2.2, the number of coarse basis functions per aggregate is not prescribed *a priori*.

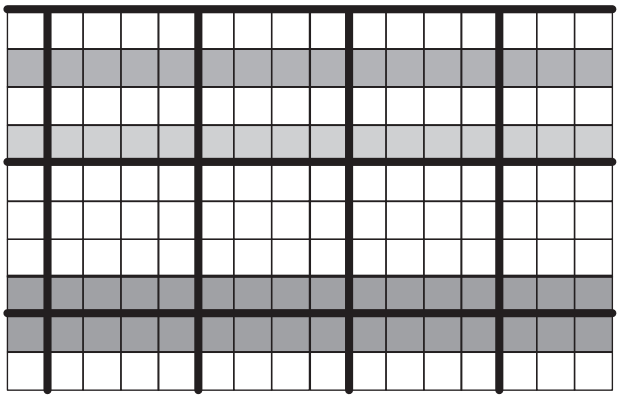

Figure 20
Medium with channels.

In multiscale methods, the number of degree of freedom per aggregate is prescribed in advance. For a scalar problem, only one coarse basis function is assigned to a coarse grid vertex. It has been explained in Section 4.1.1.1 that even for sophisticated multiscale methods, it might not be enough for channelized media with changing flow conditions. Whereas the spectral coarse space construction works well for arbitrary channels configuration, see Figure 20, as we have seen in Section 2.4.

CONCLUSION AND PROSPECTS

After having introduced Schwarz domain decomposition methods, we have presented the spectral coarse space introduced in [2] and later analyzed in [13]. It is practically optimal in the sense that a larger coarse space does not bring much improvement while a smaller one has a poor performance, see Section 2.4.1. Moreover, the method adapts automatically to the heterogeneities of the problem. If necessary, more than one coarse basis function is allowed per aggregate. This construction is supported by a theoretical condition number estimate independent of the heterogeneities of the physical problem, see Theorem 2.2. In coarse spaces built using multiscale methods, such a theorem cannot hold since only one degree of freedom is allowed per aggregate, see Section 4.2. This is why these methods have problems with channelized permeability distributions. A cure proposed [15] is to use a suitable spectral coarse space as a basis for a MsFEM method. Our DtN coarse space could be used in MsFEM methods in the same manner.

The spectral coarse space was developed, tested and analyzed in the finite element framework. In reservoir or basin simulations, finite volume discretizations are usually preferred to finite element discretizations. The extension of the spectral coarse space of Section 2 to a finite volume discretization is thus mandatory for its use in subsurface modeling. As explained in Section 2.2, the rationale behind this coarse space is written in terms of the original model *i.e.* in terms of partial differential equations. Thus the basis of the method does not depend on the discretization scheme. Therefore the definition and implementation of the spectral coarse space in a finite volume discretization will demand some work but can definitely be done. It would improve the method introduced in [25] by selecting in a sure (Theorem 2.2) and optimal (Sect. 2.4.1) manner more efficient coarse spaces when the channelized character of the permeability distribution makes it necessary.

REFERENCES

1 Toselli A., Widlund O. (2004) *Domain Decomposition Methods - Algorithms and Theory*, Vol. **34**, Springer Series in Computational Mathematics, Springer.

2 Nataf F., Xiang H., Dolean V., Spillane N. (2011) A coarse space construction based on local Dirichlet to Neumann maps, *SIAM J. Sci Comput*. **33**, 4, 1623-1642.

3 Schwarz H.A. (1870) Über einen Grenzübergang durch alternierendes Verfahren, *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich* **15**, 272-286.

4 Cai X.-C., Sarkis M. (1999) A restricted additive Schwarz preconditioner for general sparse linear systems, *SIAM Journal on Scientific Computing* **21**, 239-247.

5 Balay S., Gropp W.D., McInnes L.C., Smith B.F. (2001) PETSc users manual, Technical Report ANL95/11 -Revision 2.1.1, Argonne National Laboratory.

6 Saad Y. (1996) *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company.

7 Cai X.-C., Farhat C., Sarkis M. (1998) A minimum overlap restricted additive Schwarz preconditioner and applications to 3D flow simulations, *Contemporary Mathematics* **218**, 479-485.

8 Nataf F., Xiang H., Dolean V. (2010) A two level domain decomposition preconditioner based on local Dirichlet-to-Neumann maps, *C. R. Mathématique* **348**, 21-22, 1163-1167.

9 Japhet C., Nataf F., Roux F.-X. (1998) Extension of a coarse grid preconditioner to non-symmetric problems, in *Domain decomposition methods 10*, Mandel J., Farhal C., Cai X.-C. (eds), (Boulder, CO, 1997), Contemp. Math, **218**, 279-286. doi: 10.1090/conm/218/3019.

10 Nabben R., Vuik C. (2004) A comparison of deflation and coarse grid correction applied to porous media flow, *SIAM Journal on Numerical Analysis* **42**, 1631-1647.

11 Nicolaides R.A. (1987) Deflation of conjugate gradients with applications to boundary value problems, *SIAM J. Numer. Anal*. **24**, 2, 355-365.

12 Escobar J.F. (1997) The geometry of the first non-zero Stekloff eigenvalue, *J. Funct. Anal*. **150**, 544-556.

13 Dolean V., Nataf F., Scheichl R., Spillane N. (2012) Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet–to–Neumann maps, *Comp. Meth. Appl. Math* **12**, 4. URL http://hal.archives-ouvertes.fr/hal-00586246/.

14 Efendiev Y., Galvis J., Vassilevski P.S. (2011) Spectral element agglomerate algebraic multigrid methods for elliptic problems with high contrast coefficients, in *Domain Decomposition Methods in Science and Engineering XIX*, Huang Y., Kornhuber R., Widlund O., Xu J. (eds), Lecture Notes in Computational Science and Engineering **78**, 407-414, Springer, Berlin.

15 Efendiev Y., Galvis J., Wu X.-H. (2011) Multiscale finite element methods for high-contrast problems using local spectral basis functions, *Journal of Computational Physics* **230**, 937-955.

16 Galvis J., Efendiev Y. (2010) Domain decomposition preconditioners for multiscale flows in high-contrast media, *Multiscale Model. Simul.* **8**, 4, 1461-1483.

17 Galvis J., Efendiev Y. (2010) Domain decomposition preconditioners for multiscale flows in high contrast media: Reduced dimension coarse spaces, *Multiscale Model. Simul.* **8**, 5, 1621-1644.

18 Hecht F. (2010) *FreeFem++*. Numerical Mathematics and Scientific Computation, Laboratoire J.L. Lions, Université Pierre et Marie Curie, http://www.freefem.org/ff++/, 3.7 edition.

19 Karypis G., Kumar V. (1998) METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, *Technical report*, Department of Computer Science, University of Minnesota. http://glaros.dtc.umn.edu/gkhome/views/metis.

20 Graham I.G., Scheichl R. (2007) Robust domain decomposition algorithms for multiscale PDEs, *Numerical Methods for Partial Differential Equations* **23**, 4, 859-878.

21 Spillane N., Dolean V., Hauret P., Nataf F., Pechstein C., Scheichl R. (2011) A robust two level domain decomposition preconditioner for systems of PDEs, *Comptes Rendus Mathématique* **349**, 23-24, 1255-1259.

22 Efendiev Y., Hou T.Y. (2009) *Multiscale finite element methods,* Volume 4 of *Surveys and Tutorials in the Applied Mathematical Sciences*, Springer, New York. ISBN 978-0-387-09495-3. Theory and applications.

23 Hesse M.A., Mallison B.T., Tchelepi H.A. (2008) Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations, *Multiscale Model. Simul.* **7**, 2, 934-962. ISSN 1540-3459. doi: 10.1137/070705015.

24 Nordbotten J.M., Bjørstad P.E. (2008) On the relationship between the multiscale finite-volume method and domain decomposition preconditioners, *Comput. Geosci.* **12**, 3, 367-376. ISSN 1420-0597. doi: 10.1007/s10596-007-9066-6.

25 Zhou H., Tchelepi H.A. (2011) Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models, *SPE Reservoir Simulation Symposium*, The Woodlands, Texas, USA, 21-22 Feb. *SPE paper* 141473-MS.

26 Chen Y., Durlofsky L.J., Gerritsen M., Wen X.H. (2003) A coupled local-global upscaling approach for simulating flow in highly heterogeneous formations, *Advances in Water Resources* **26**, 10, 1041-1060. ISSN 0309-1708. doi: 10.1016/S0309-1708(03)00101-5.

27 Chen Yuguang., Durlofsky L.J. (2006) Efficient incorporation of global effects in upscaled models of two-phase flow and transport in heterogeneous formations, *Multiscale Model. Simul.* **5**, 2, 445-475. ISSN 1540-3459. doi: 10.1137/060650404.

28 Durlofsky L.J., Efendiev Y., Ginting V. (2007) An adaptive local-global multi-scale finite volume element method for two-phase flow simulations, *Advances in Water Resources* **30**, 3, 576-588. ISSN 0309-1708. doi: 10.1016/j.advwatres.2006.04.002.

29 Vuik C., Segal A., Meijerink J.A. (1999) An effcient preconditioned cg method for the solution of a class of layered problems with extreme contrasts in the coefficients, *J. Comput. Phys.* **152**, 385-403.

30 Aarnes J., Hou T.Y. (2002) Multiscale domain decomposition methods for elliptic problems with high aspect ratios, *Acta Math. Appl. Sin. Engl. Ser.* **18**, 1, 63-76. ISSN 0168-9673. doi: 10.1007/s102550200004.

Manuscript accepted in November 2013
Published online in July 2014