

## A COARSE SPACE CONSTRUCTION BASED ON LOCAL DIRICHLET-TO-NEUMANN MAPS\*

FRÉDÉRIC NATAF<sup>†</sup>, HUA XIANG<sup>‡</sup>, VICTORITA DOLEAN<sup>§</sup>, AND NICOLE SPILLANE<sup>†</sup>

**Abstract.** Coarse-grid correction is a key ingredient of scalable domain decomposition methods. In this work we construct coarse-grid space using the low-frequency modes of the subdomain Dirichlet-to-Neumann maps and apply the obtained two-level preconditioners to the extended or the original linear system arising from an overlapping domain decomposition. Our method is suitable for parallel implementation, and its efficiency is demonstrated by numerical examples on problems with large heterogeneities for both manual and automatic partitionings.

**Key words.** domain decomposition, coarse grid, deflation, heterogeneous coefficients

**AMS subject classifications.** 65N55, 65F10, 65N30

**DOI.** 10.1137/100796376

### NOTATION.

$A$	coefficient matrix of the linear system $Ax = b$
$M$	preconditioner for $A$
$Z, Y$	full rank matrices which span the coarse-grid subspaces
$E$	$E = Y^T AZ$ , Galerkin matrix or coarse-grid matrix
$\Xi$	$\Xi = ZE^{-1}Y^T$ , coarse-grid correction matrix in multigrid and domain decomposition methods
$P_D$	$P_D = I - A\Xi = I - AZ(Y^T AZ)^{-1}Y^T$
$Q_D$	$Q_D = I - \Xi A = I - Z(Y^T AZ)^{-1}Y^T A$
$P_{BNN}$	$P_{BNN} = Q_D M^{-1} P_D + ZE^{-1}Y^T$
$P_{ADEF2}$	$P_{ADEF2} = Q_D M^{-1} + ZE^{-1}Z^T$

**1. Introduction.** We consider the solution of the linear system  $Ax = b \in \mathbb{R}^p$  arising from the discretization of an elliptic boundary value problem (BVP) of the type

$$(1.1) \quad \begin{aligned} \eta u - \operatorname{div}(\kappa \nabla u) &= f && \text{in } \Omega, \\ B(u) &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where  $\Omega$  is a bounded domain of  $\mathbb{R}^d$  ( $d = 2$  or  $3$ ),  $B$  is a suitable boundary condition,  $\kappa$  is a positive diffusion function which can be discontinuous, and  $\eta \geq 0$ . When using an iterative method in a one-level domain decomposition framework, one may encounter a long stagnation or a slow convergence, especially when the number of subdomains is

---

\*Submitted to the journal's Methods and Algorithms for Scientific Computing section May 25, 2010; accepted for publication (in revised form) April 26, 2011; published electronically July 21, 2011. The first and second authors acknowledge the support of the French National Research Agency (ANR PETAL) and CNRS via the GDR MOMAS.

<http://www.siam.org/journals/sisc/33-4/79637.html>

<sup>†</sup>Laboratoire J.L. Lions, CNRS UMR 7598, Université Pierre et Marie Curie, Paris 75005, France (nataf@ann.jussieu.fr, spillane@ann.jussieu.fr).

<sup>‡</sup>School of Mathematics and Statistics, Wuhan University, Wuhan 430072, P. R. China (xiang@ann.jussieu.fr).

<sup>§</sup>Laboratoire J.-A. Dieudonné, CNRS UMR 6621, Université de Nice-Sophia Antipolis, Parc Valrose, 06108 Nice Cedex 02, France (Victorita.Dolean@unice.fr).

large. Even when  $\kappa \equiv 1$ , the convergence of a one-level domain decomposition method (DDM) presents a long plateau of stagnation. Its length is related to the number of subdomains of the decomposition in one direction. For example, we know that for the problem divided into  $N$  subdomains in a one-way partitioning, convergence can never be achieved in less than  $N - 1$  iterations since the exchange of information between the subdomains is reduced to their common interfaces. Thus, the global information transfers from one subdomain only to its neighbors [13, 16]. One needs a two-level method to have a scalable algorithm, i.e., an algorithm whose convergence rate is weakly dependent on the number of subdomains; see [26] and references therein.

Two-level domain decomposition methods are closely related to multigrid (MG) methods and deflation corrections; see [24] and references therein. These methods are defined by two ingredients: a full rank matrix  $Z \in \mathbb{R}^{p \times m}$  with  $m \ll p$  and an algebraic formulation of the correction. These techniques imply solving a reduced-size problem of order  $m \times m$ , called a coarse-grid problem. The space spanned by the columns of  $Z$  should ideally contain the vectors responsible for the stagnation of the one-level method. We will come back later to the choice of  $Z$  in the framework of DDMs and focus for now on the various algebraic ways to improve convergence by using a coarse grid.

According to [24], for a DDM, a well-known coarse-grid correction preconditioner is of the form  $I + ZE^{-1}Z^T$ , where  $E = Z^T AZ$  is the coarse-grid matrix used to speed up convergence. The abstract additive coarse-grid correction proposed in [18] is  $M^{-1} + ZE^{-1}Z^T$ , where  $M^{-1}$  is the additive Schwarz preconditioner, a sum of local solvers in each subdomain, which can be implemented in parallel. The first term is a fine grid solver, and the second term represents a coarse solver. Hence it is called the two-level additive Schwarz preconditioner. The BPS preconditioner introduced by Bramble, Pasciak, and Schatz [1] is of this type. In the context of DDMs, we mention the balancing Neumann–Neumann preconditioner and the FETI (finite element tearing interconnection) algorithm, which have been extensively investigated; see [26] and references therein. For symmetric systems the balancing preconditioner was proposed by Mandel [14]. The abstract balancing preconditioner [14] for nonsymmetric systems reads [8]

$$(1.2) \quad P_{BNN} = Q_D M^{-1} P_D + ZE^{-1}Y^T.$$

For the preconditioner  $P_{BNN}$ , if we choose the initial approximation  $x_0 = \Xi b$ , then the action of  $P_D$  is not required in practice; see [26, p. 48]. Note that the MG V(1,1)-cycle preconditioner  $P_{MG}$  is closely related to  $P_{BNN}$ . Choosing the proper smoother  $M$  in  $P_{BNN}$ , we can ensure that  $P_{MG}$  and  $P_{BNN}$  are symmetric positive definite (SPD), and  $P_{MG}A$  and  $P_{BNN}A$  have the same spectrum [25].

For an SPD system, by choosing  $Y = Z$ , the authors in [24] define

$$(1.3) \quad P_{ADEF2} = Q_D M^{-1} + ZE^{-1}Z^T,$$

which is as robust as  $P_{BNN}$  but usually less expensive [24]. The two-level hybrid Schwarz preconditioner in [23, p. 48] has the same form as  $P_{ADEF2}$ . It is shown in [24] that with a proper choice of the starting vector the preconditioners  $P_{ADEF2}$  and  $P_{BNN}$  are equivalent.

Given the coarse-grid subspace, we can construct the two-level preconditioners above. An effective two-level preconditioner is highly dependent on the choice of the coarse-grid subspace. We will now focus on the choice of the coarse space  $Z$  in the context of DDMs for problems of type (1.1) with heterogeneous coefficients. In this case, the coarse space is made of vectors with support in one subdomain

or one subdomain and its neighbors. When the jumps in the coefficients are inside the subdomains (and not on the interface) or across the interface separating the subregions, the use of a fixed number of vectors per subdomain in  $Z$  gives good results; see [6, 15, 19, 20, 4, 5]. When the discontinuities are along the interfaces between the subdomains, results are not so good.

Here, we propose constructing a coarse subspace such that the two-level method is robust with respect to heterogeneous coefficients for an arbitrary domain decomposition. Even if the discontinuities are along the interface, as in the case with layered coefficients and a one-way decomposition in the orthogonal direction (see section 4.1), the iteration counts are very stable with respect to jumps of the coefficients. Our method is based on the low-frequency modes associated with the Dirichlet-to-Neumann (DtN) map on each subdomain. After obtaining the eigenvectors associated with the small eigenvalues of DtN, we use the harmonic extension to the whole subdomain. It is efficient even for problems with large discontinuities in the coefficients. Moreover, it is suitable for parallel implementation. We apply such a two-level preconditioner to the original linear system (2.1) and to the extended one (2.2) arising from the DDM.

The paper is organized as follows. In section 2, we introduce the two-level preconditioners: the additive Schwarz (AS), the restricted additive Schwarz (RAS), and the Jacobi–Schwarz (JS) with the coarse-grid correction. The construction of coarse-grid spaces is presented in section 3. In section 4, numerical tests on the model problem demonstrate the efficiency of our method. Some concluding remarks are given in section 5.

**2. Algebraic DDMs.** Without loss of generality, we consider here a decomposition of a domain  $\Omega$  into two overlapping subdomains  $\Omega_1$  and  $\Omega_2$ . The overlap is denoted by  $O := \Omega_1 \cap \Omega_2$ . This yields a partition of the domain:  $\bar{\Omega} = \bar{\Omega}_I^{(1)} \cup \bar{O} \cup \bar{\Omega}_I^{(2)}$ , where  $\Omega_I^{(i)} := \Omega_i \setminus \bar{O}$ ,  $i = 1, 2$ . At the algebraic level this corresponds to a partition of the set of indices  $\mathcal{N}$  into three sets:  $\mathcal{N}_I^{(1)}$ ,  $\mathcal{N}_O$ , and  $\mathcal{N}_I^{(2)}$ ; see Figure 2.1.

After the discretization of the BVP (1.1) defined in domain  $\Omega$ , we obtain a linear system of the following form:

$$(2.1) \quad A u := \begin{bmatrix} A_{II}^{(1)} & A_{IO}^{(1)} \\ A_{OI}^{(1)} & A_{OO} & A_{OI}^{(2)} \\ & A_{IO}^{(2)} & A_{II}^{(2)} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_O \\ u_I^{(2)} \end{bmatrix} = \begin{bmatrix} f_I^{(1)} \\ f_O \\ f_I^{(2)} \end{bmatrix}.$$

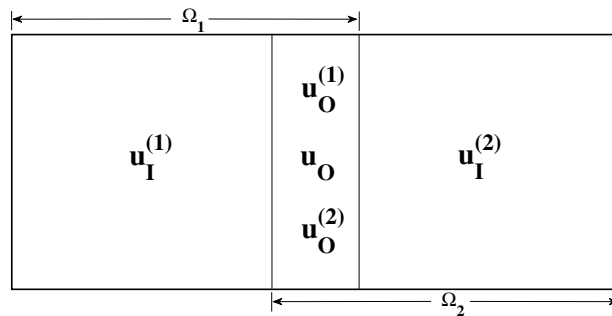


FIG. 2.1. Decomposition into two overlapping subdomains.

We can also define the extended linear system by considering twice the variables located in the overlapping region,

$$(2.2) \quad \tilde{A} \tilde{u} := \begin{bmatrix} A_{II}^{(1)} & A_{IO}^{(1)} & & & \\ A_{OI}^{(1)} & A_{OO} & & A_{OI}^{(2)} & \\ A_{OI}^{(1)} & & A_{OO} & A_{OI}^{(2)} & \\ & & & A_{OI}^{(2)} & A_{II}^{(2)} \\ & & & & & u_I^{(2)} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_O^{(1)} \\ u_O^{(2)} \\ u_I^{(2)} \end{bmatrix} = \begin{bmatrix} f_I^{(1)} \\ f_O \\ f_O \\ f_I^{(2)} \end{bmatrix},$$

where the subscript “O” stands for “overlap,”  $u_O^{(i)}$  are the duplicated variables in the overlapping domain  $O$ , and  $u_I^{(i)}$  are variables in the subdomain  $\Omega_i$ . It is easy to check that if  $A_{OO}$  is invertible, there is an equivalence between problems (2.1) and (2.2).

Classical preconditioners for problem (2.1) are the AS and the RAS methods; see [2] or [26] and references therein. Let  $R_j$  be the rectangular restriction matrix to subdomain  $\Omega_j$ ,  $j = 1, 2$ . Let  $D_j$ ,  $j = 1, 2$ , be diagonal matrices which correspond to a partition of unity in the sense that

$$R_1^T D_1 R_1 + R_2^T D_2 R_2 = I.$$

By defining  $\tilde{R}_j := D_j R_j$ , we have

$$\tilde{R}_1^T R_1 + \tilde{R}_2^T R_2 = I.$$

Then the AS preconditioner reads

$$(2.3) \quad M_{AS}^{-1} := R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2,$$

and the RAS method reads

$$(2.4) \quad u^{n+1} = u^n + (\tilde{R}_1^T A_1^{-1} R_1 + \tilde{R}_2^T A_2^{-1} R_2)(f - Au^n),$$

where  $A_i := R_i A R_i^T$ ,  $i = 1, 2$ . From the iterative scheme (2.4), we can define the preconditioner

$$M_{RAS}^{-1} := \tilde{R}_1^T A_1^{-1} R_1 + \tilde{R}_2^T A_2^{-1} R_2.$$

Note that the RAS preconditioner is nonsymmetric. It leads to an iterative method that is identical to the discretization of the continuous JS method; see [7].

An illustration of the RAS method is given in Figure 2.2 for a three-subdomain decomposition, where  $\Omega_i$  are overlapping subdomains and  $\Omega_i^*$  are nonoverlapping subdomains. If we take  $D_i$  to have zero entries for nodes in the region  $\Omega_i \setminus \Omega_i^*$ , we define

$$\begin{aligned} \tilde{R}_1^T &= \begin{bmatrix} I_{\Omega_1^*} \\ 0 \\ 0 \end{bmatrix}, & \tilde{R}_2^T &= \begin{bmatrix} 0 \\ I_{\Omega_2^*} \\ 0 \end{bmatrix}, & \tilde{R}_3^T &= \begin{bmatrix} 0 \\ 0 \\ I_{\Omega_3^*} \end{bmatrix}, \\ R_1 &= [I_{\Omega_1} \quad 0 \quad 0], & R_2 &= [0 \quad I_{\Omega_2} \quad 0], & R_3 &= [0 \quad 0 \quad I_{\Omega_3}], \end{aligned}$$

where  $I_{\Omega_i}$  is the identity matrix whose dimension equals the number of unknowns in  $\Omega_i$ .

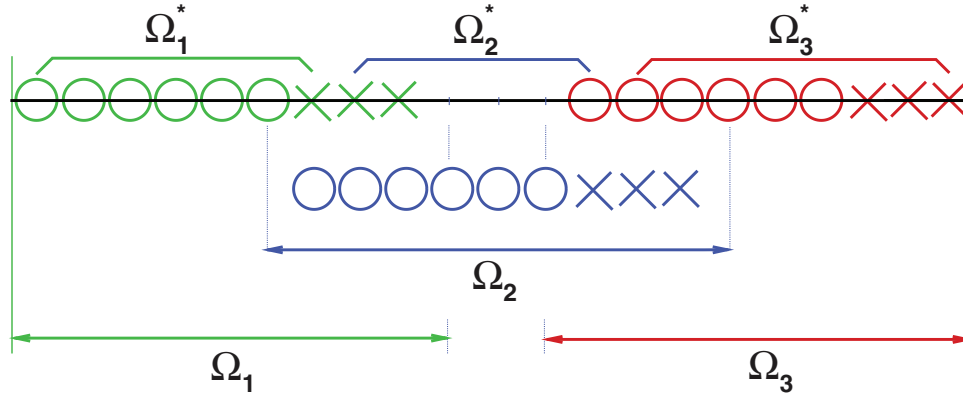


FIG. 2.2. RAS ( $n_{domain} = 3$ ,  $n_{overlap} = 4 - 1$ ,  $n_{xpt} = 6$ ; see section 4.1 for these parameters). The circles and crosses stand for the unknowns in each subdomain.

For our extended linear system (2.2), a natural preconditioner can be built by using its diagonal blocks. For the two-subdomain case (2.2), the block diagonal preconditioner  $M_{JS}$  is

$$(2.5) \quad M_{JS} := \begin{bmatrix} A_{II}^{(1)} & A_{IO}^{(1)} & & & & \\ A_{OI}^{(1)} & A_{OO} & & & & \\ & & A_{OO} & A_{OI}^{(2)} & & \\ & & A_{IO}^{(2)} & A_{II}^{(2)} & & \end{bmatrix},$$

and one can easily notice that  $M_{JS}^{-1}$  can be computed in parallel. The resulting method will be referred to as the JS method. When used in a Richardson algorithm such as in (2.4), it was proved in [3] that  $M_{JS}$  applied to (2.2) and  $M_{RAS}$  applied to (2.1) lead to equivalent algorithms. But, as we shall see from numerical experiments, two-level methods applied to (2.2) or to (2.1) are not necessarily equivalent.

Note that even though the original matrix  $A$  is symmetric, the extended one  $\tilde{A}$  is not. As for the preconditioners,  $M_{AS}$  and  $M_{JS}$  are symmetric but  $M_{RAS}$  is not. As a result, the only case where we have both a symmetric matrix and a symmetric preconditioner is when  $M_{AS}$  applies to the original matrix  $A$ . In this case, the Krylov method that we use is the CG algorithm. In the other two cases (namely,  $M_{JS}$  applied to the extended matrix and  $M_{RAS}$  applied to the original system), we use GMRES [22].

Using preconditioners  $M_{AS}$ ,  $M_{JS}$ , or  $M_{RAS}$ , we can remove the influence of very large eigenvalues of the coefficient matrix, which correspond to high-frequency modes. 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.

**3. The coarse-grid space construction.** A key problem is the choice of the coarse subspace. Ideally, we can choose the deflation subspace  $Z$  which consists of the eigenvectors associated with the small eigenvalues of the preconditioned operator. But the lower part of the spectrum of a matrix is costly to obtain. The cost is in any case larger than the cost of solving a linear system. Thus, there is a need to

choose the coarse space a priori. For instance, in [17], Nicolaidis defines the deflation subspace  $Z$  as

$$(z_j)_i = \begin{cases} 1 & \text{if } i \in \Omega_j^*, \\ 0 & \text{otherwise,} \end{cases}$$

whose matrix form is

$$(3.1) \quad Z = \begin{bmatrix} 1_{\Omega_1^*} & 0 & 0 \\ 0 & 1_{\Omega_2^*} & 0 \\ 0 & 0 & 1_{\Omega_3^*} \end{bmatrix},$$

where  $1_{\Omega_i^*}$  is a vector of all ones, and its length equals the number of unknowns in  $\Omega_i^*$ . Recall that the subdomains  $\Omega_i^*$  are nonoverlapping as shown in Figure 2.2.

In [27, 28], the projection vectors  $z_i$  are chosen in a similar but more complicated way. Definition (3.1) is also used as the aggregation-based coarse level operator in [9]. Originally (3.1) was used for disjoint sets but not for the overlapping case. In the following, we use it in the overlapping case as well. This coarse space performs well in the constant coefficient case. But when there are jumps in the coefficients, it cannot prevent stagnation in the convergence.

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 and arbitrary domain decompositions. For both the original system (2.1) and the extended one (2.2), we still choose  $Z$  such that it has the form

$$(3.2) \quad Z = \begin{bmatrix} W^1 & 0 & \cdots & 0 \\ \vdots & W^2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & W^N \end{bmatrix},$$

where  $N$  is the number of overlapping subdomains. However,  $W^i$  is now a rectangular matrix whose columns are based on the harmonic extensions of the eigenvectors corresponding to small eigenvalues of the DtN map in each subdomain  $\Omega_i$ . Note 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 DtN map  $\text{DtN}_{\Omega_i}$ . Let  $u : \Gamma_i \mapsto \mathbb{R}$ ,

$$\text{DtN}_{\Omega_i}(u) = \kappa \frac{\partial v}{\partial n_i} \Big|_{\Gamma_i},$$

where  $v$  satisfies

$$(3.3) \quad \begin{cases} \mathcal{L}(v) := (\eta - \text{div}(\kappa \nabla))v = 0 & \text{in } \Omega_i, \\ v = u & \text{on } \Gamma_i, \end{cases}$$

and  $\Gamma_i$  is the interface boundary. If the subdomain is not a floating one (i.e.,  $\partial\Omega_i \cap \partial\Omega \neq \emptyset$ ), 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:

$$(3.4) \quad \text{DtN}_{\Omega_i}(u) = \lambda \kappa u$$

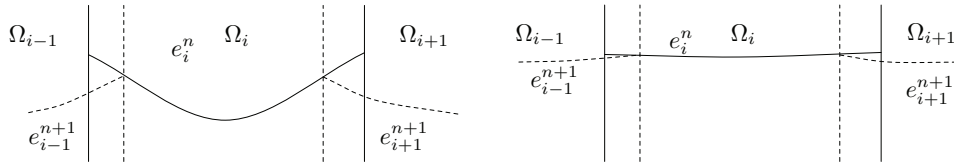


FIG. 3.1. Fast (left) or slow (right) convergence of the Schwarz algorithm.

with

$$(3.5) \quad \lambda < 1/\text{diam}(\Omega_i),$$

where  $\text{diam}(\Omega_i)$  is the diameter of subdomain  $\Omega_i$ .

We first motivate our choice of a coarse space based on the DtN map. We write the original Schwarz method at the continuous level, where the domain  $\Omega$  is decomposed in a one-way partitioning. 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  $\Omega_i$  at step  $n$  of the algorithm satisfies

$$\begin{aligned} \mathcal{L}(e_i^{n+1}) &= 0 && \text{in } \Omega_i, \\ e_i^{n+1} &= \sum_{j \neq i} \xi_i^j e_j^n && \text{on } \bar{\Omega}_i \cap \partial\Omega_j, \end{aligned}$$

where  $\xi_i^j$  is such that

$$\sum_{j \neq i} \xi_i^j = 1_{\partial\Omega_i}.$$

In the 1D (one-dimensional) example sketched in Figure 3.1, 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 (BC). Indeed, a small value for this BC 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 keep only eigenvectors with eigenvalues smaller than  $1/\text{diam}(\Omega_i)$  in the coarse space. We start with the constant coefficient case  $\kappa = 1$ . In this case, the smallest eigenvalue of the DtN map is zero and corresponds to the constant function 1. For a shape regular subdomain, the first positive eigenvalue is of order  $1/\text{diam}(\Omega_i)$ ; see [10]. Keeping only the constant function 1 in the coarse space leads to good numerical convergence; see Figure 4.2 below. In the case of high contrasts in the coefficient  $\kappa$ , the smallest eigenvalue of the DtN map is still zero. But due to the variation of the coefficients we may have positive eigenvalues smaller than  $1/\text{diam}(\Omega_i)$ . In order to have convergence behavior similar to that of the constant coefficient case, it is natural to keep all eigenvectors with eigenvalues smaller than  $1/\text{diam}(\Omega_i)$ .

To obtain the discrete form of the DtN map, we consider the variational form of

(3.3). Let's define the bilinear form  $a_i : H^1(\Omega_i) \times H^1(\Omega_i) \rightarrow \mathbb{R}$ ,

$$a_i(w, v) := \int_{\Omega_i} \eta w v + \kappa \nabla w \cdot \nabla v.$$

With a finite element basis  $\{\phi_k\}$ , the coefficient matrix of a Neumann BVP in domain  $\Omega_i$  is

$$A_{kl}^{(i)} = \int_{\Omega_i} \eta \phi_k \phi_l + \kappa \nabla \phi_k \cdot \nabla \phi_l.$$

Let  $I$  (resp.,  $\Gamma_i$ ) 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  $\Omega$  the coefficient matrix is given by

$$A_{kl} = \int_{\Omega} \eta \phi_k \phi_l + \kappa \nabla \phi_k \cdot \nabla \phi_l.$$

With block notation, we have

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

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 (3.3), we use Dirichlet BCs. Let  $U \in \mathbb{R}^{n_{\Gamma_i}}$  and  $u := \sum_{k \in \Gamma_i} U_k \phi_k$ . Let  $v := \sum_{k \in I} V_k \phi_k + \sum_{l \in \Gamma_i} V_l \phi_l$  be the finite element approximation of the solution of (3.3). Letting  $V_I = (V_k)_{k \in I}$ , we have with obvious notation

$$(3.6) \quad A_{II} V_I + A_{I\Gamma_i} U = 0.$$

A variational definition of the flux reads

$$\int_{\Gamma_i} \kappa \frac{\partial v}{\partial n} \phi_k = \int_{\Omega_i} \eta v \phi_k + \kappa \nabla v \cdot \nabla \phi_k$$

for all  $\phi_k, k \in \Gamma_i$ . So the variational formulation of the eigenvalue problem (3.4) reads

$$(3.7) \quad \int_{\Omega_i} \eta v \phi_k + \kappa \nabla v \cdot \nabla \phi_k = \lambda \int_{\Gamma_i} \kappa v \phi_k$$

for all  $\phi_k, k \in \Gamma_i$ . Let  $M_{\kappa, \Gamma_i}$  be the weighted mass matrix

$$(M_{\kappa, \Gamma_i})_{kl} := \int_{\Gamma_i} \kappa \phi_k \phi_l \quad \forall k, l \in \Gamma_i.$$

The compact form of (3.7) is

$$A_{\Gamma_i \Gamma_i}^{(i)} U + A_{\Gamma_i I} V_I = \lambda M_{\kappa, \Gamma_i} U.$$

With (3.6), the discrete form of (3.4) is a generalized eigenvalue problem

$$(3.8) \quad (A_{\Gamma_i \Gamma_i}^{(i)} - A_{\Gamma_i I} A_{II}^{-1} A_{I\Gamma_i}) U = \lambda M_{\kappa, \Gamma_i} U.$$

The step-by-step procedure for constructing the rectangular matrices  $W^i$  in the coarse space matrix  $Z$  (see (3.2)) is as follows.

ALGORITHM 1. *In parallel for all subdomains  $1 \leq i \leq N$ , the following hold:*



1. Compute eigenpairs of (3.8),  $(U_1^i, \lambda_1^i), (U_2^i, \lambda_2^i), \dots, (U_{m_i}^i, \lambda_{m_i}^i)$ , such that

$$\lambda_1^i \leq \dots \leq \lambda_{m_i}^i < 1/\text{diam}(\Omega_i) \leq \lambda_{m_i+1}^i \leq \dots$$

2. Compute their harmonic extensions,

$$V_k^i := (-A_{II}^{-1} A_{I\Gamma_i} U_k^i, U_k^i)^T, \quad 1 \leq k \leq m_i.$$

3. The final formula for the rectangular matrix  $W^i$  with  $m_i$  columns depends on the system to be solved:

If the extended system is solved (see (2.2)), then

$$W^i = [V_1^i | \dots | V_{m_i}^i].$$

If the original system is solved (see (2.1)), then

$$W^i = [D_i V_1^i | \dots | D_i V_{m_i}^i].$$

We call this procedure the  $Z_{D2N}$  method. We also use  $Z_{D2N}$  to denote the coarse-grid space constructed by this method. Its construction is fully parallel. Similarly, we call  $Z_{Nico}$  the method of Nicolaides or the corresponding coarse-grid space. Let us remark that when  $\eta = 0$  and the subdomain does not touch the boundary of  $\Omega$ , 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 vector per subdomain.

Note that when we work on the original system, the definition of the coarse space involves a partition of unity via the matrices  $D_i$ . Thus the vectors of the coarse space are able to span the zero energy modes of the original problem if there are no Dirichlet BCs. In the following numerical tests, we construct the diagonal matrices  $D_i$  in the spirit of the RAS method. Let  $f_i : \Omega_i \mapsto \mathbb{R}$  be the characteristic function of  $\bar{\Omega}_i^*$ . We define  $\chi_i(x)$  as the P1 finite element interpolation of  $f_i(x)/\sum_j f_j(x)$ . Then, we set (with MATLAB notation)  $D_i := \text{diag}(\chi_i)$ .

**4. Numerical results.** From a practical point of view, it is more difficult to work on the extended system (2.2) than on the original one (2.1). Indeed, when working on (2.2), we have to create extra data structures. For this reason, we first consider in section 4.1 a one-way decomposition of the domain. This enables us to work both on the original system (2.1) and on the extended one (2.2). In section 4.2 an arbitrary decomposition is considered; we will work only on the original system (2.1). In all numerical results, the residual reduction factor to convergence is  $10^{-6}$ .

**4.1. One-way domain decomposition.** In order to illustrate numerically the behavior of the coarse space, we first consider the following model problem:

$$\mathcal{L}u := \left( -\frac{\partial}{\partial x} c(y) \frac{\partial}{\partial x} - \frac{\partial}{\partial y} d(y) \frac{\partial}{\partial y} + \eta(y) \right) u(x, y) = f(x, y) \quad \text{in } (0, L) \times (0, 1),$$

where the length  $L$  of the domain is proportional to the number of subdomains.

We use Neumann BCs on the top and the bottom, a Dirichlet BC on the left, and  $(\partial/\partial n + \alpha)u = 0, \alpha \ll 1$ , on the right boundary; see Figure 4.1.

All the test cases are described in Table 4.1. We first consider the constant coefficient case with  $c(y) = d(y) = 1.0$  (CONST in Table 4.1). Afterward we will

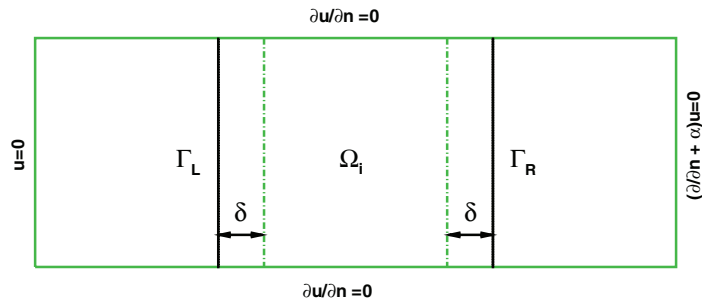


FIG. 4.1. Subdomains and boundary conditions.

TABLE 4.1  
Test cases:  $b = 1$ ,  $a = 100000$ .

Case	Remarks
CONST	$c(y) = d(y) = 1.0$ , constant coefficients
HPL1	$c(y) = d(y) = \text{val}[10 * y]$ , $\text{val} = [b \ b \ b \ a \ a \ a \ b \ b \ b]$
HPL2	$c(y) = d(y) = \text{val}[10 * y]$ , $\text{val} = [b \ b \ a \ a \ b \ b \ a \ a \ b \ b]$
HPL3	$c(y) = d(y) = \text{val}[10 * y]$ , $\text{val} = [a \ a \ b \ b \ a \ a \ b \ b \ a \ a]$
HPL4	$c(y) = d(y) = \text{val}[10 * y]$ , $\text{val} = [a \ b \ b \ a \ b \ b \ a \ b \ b \ a]$
HPL5	$c(y) = d(y) = \text{val}[10 * y]$ , $\text{val} = [b \ a \ b \ a \ b \ a \ b \ a \ b \ a]$

test more difficult cases, with discontinuities in the coefficients, such as  $c(y) = d(y) = \text{val}[10 * y]$ , where  $\text{val}$  is an array that defines the heterogeneity pattern and that depends on two parameters  $a$  and  $b$  (e.g., in the case HPL3,  $\text{val} = [a \ a \ b \ b \ a \ a \ b \ b \ a \ a]$  means that there are three high-permeability layers). In our tests we take  $a = 10^5$ ,  $b = 1$ , and  $\eta = 10^{-7}$ .

In addition, we use the following parameters in our tests:

- $n_{\text{domain}}$  represents the number of subdomains.
- $n_{\text{overlap}}$  stands for the overlap in the  $x$ -direction; correspondingly the width of the overlap is  $\delta = (n_{\text{overlap}} + 1)h$ , where  $h$  is the mesh size.
- $(n_{\text{xpt}} + n_{\text{overlap}})$  is the number of grid points in the  $x$ -direction in one subdomain.
- $n_{\text{ypt}}$  is the number of grid points in the  $y$ -direction.

These numerical tests are performed in MATLAB. We compare the two coarse grids, namely the Nicolaides one (3.1) and the one defined in Algorithm 1 (D2N), only for the RAS and JS preconditioners. See section 4.2 for the RAS and AS methods. We use full GMRES as the linear solver, together with the left preconditioner  $P_{\text{ADEF2}}$ , which is constructed by taking  $Z$  equal to  $Z_{\text{Nico}}$  or to  $Z_{\text{D2N}}$ , where  $Z_{\text{D2N}}$  is of the form (3.2). Please note that when using the left-preconditioned GMRES on  $Ax = b$ , the residual  $\|b - A\bar{x}\|$  is returned for the approximate solution  $\bar{x}$  (see Figures 4.2–4.6).

**4.1.1. The original vs. extended system.** We first compare the effect of the coarse-grid correction when applied to the original system (2.1) (RAS method) or to the extended system (2.2) (JS method). We begin with the Nicolaides coarse-grid space. As expected when there is no coarse-grid correction, RAS and JS perform similarly and have a plateau in the convergence curve.

For the constant case in Figure 4.2, the piecewise coarse-grid space of Nicolaides is quite good for both the extended system and the original system. It works better

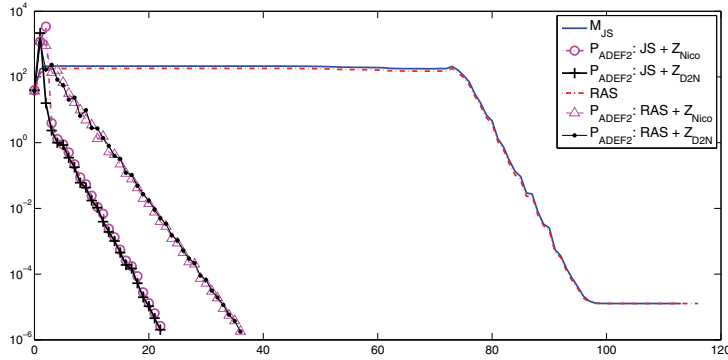


FIG. 4.2. Case CONST for JS and RAS using coarse grids.  $n_{domain} = 32$ ,  $n_{overlap} = 1$ ,  $n_{xpt} = 8$ ,  $n_{ypt} = 16$ . Both  $Z_{Nico}$  and  $Z_{D2N}$  work well for the constant case.

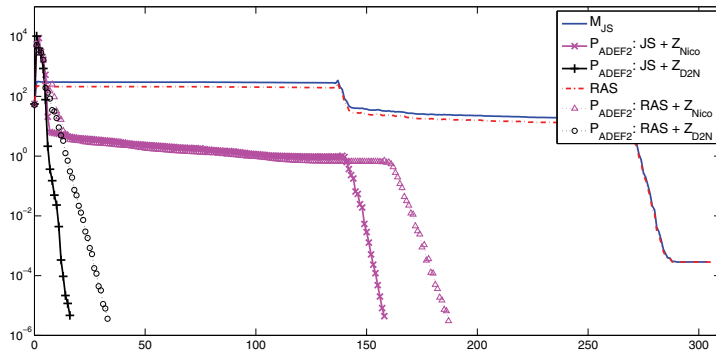


FIG. 4.3. Case HPL2 for JS and RAS with coarse-grid correction.  $n_{domain} = 64$ ,  $n_{overlap} = 1$ ,  $n_{xpt} = 8$ ,  $n_{ypt} = 16$ .  $Z_{Nico}$  does not work well, while  $Z_{D2N}$  gives fast convergence.

on the extended system (2.2) (JS method) than on the original system (2.1) using RAS (22 versus 36 iterations).

For the case with discontinuities,  $Z_{Nico}$  is not good. For this kind of problem,  $Z_{D2N}$  gives a much faster convergence; see Figure 4.3. Note that for the case HPL2, the number of small eigenvalues determined by (3.5) is 2, which is equal to the number of high-permeability layers; see [27].

We can see that our  $Z_{D2N}$  method works well on both the extended system and the original system, but it gives better convergence on the extended system. It seems to us that this is due to the fact that for JS we do not need a partition of unity. For RAS this partition of unity is necessary for the method to be able to span the zero energy modes of the original problem. But then, the local harmonicity of the coarse space components is lost. This is the reason why we will further investigate our  $Z_{D2N}$  method on the extended system.

**4.1.2. The robustness of the  $Z_{D2N}$  method.** In the following we test the robustness of the approach with respect to the various parameters of the problem: jumps of the coefficients, number of subdomains, mesh size, and size of the overlap.

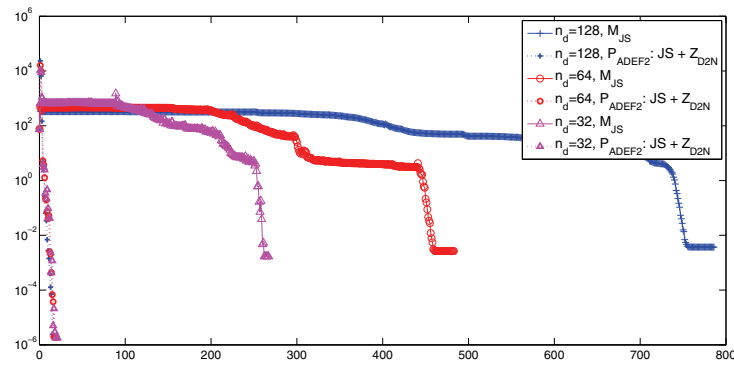


FIG. 4.4. Case HPL3 with various number of subdomains.  $n_{domain} = 128$  (crosses), 64 (circles), 32 (triangles),  $n_{overlap} = 1$ ,  $n_{ypt} = 16$ .  $Z_{D2N}$ : dotted lines, JS: solid lines.

TABLE 4.2

Case HPL2 with the  $Z_{D2N}$  coarse space for jumps in the coefficients ranging from 1 to  $10^6$ .  $n_{domain} = 32$ ,  $n_{overlap} = 3$ ,  $n_{xpt} = 8$ ,  $n_{ypt} = 16$ .

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

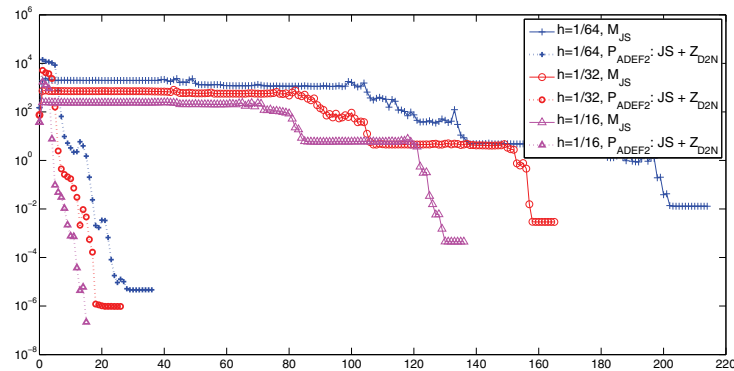


FIG. 4.5. Case HPL3 with various mesh sizes: 1/64 (crosses), 1/32 (circles), and 1/16 (triangles);  $n_{overlap} = 1$ ,  $n_{domain} = 16$ .  $Z_{D2N}$ : dotted lines, JS: solid lines.

Figure 4.4 shows that, when we use a coarse space, the iteration numbers are almost constant as the number of subdomains increases. The three convergence curves with  $Z_{D2N}$  are difficult to distinguish.

We also consider the robustness with respect to the size of the jumps. We take as an example in Table 4.2 the case HPL2 with  $a$  ranging from 1 to  $10^6$ . Using the criterion (3.5), in each subdomain two small eigenvectors are chosen to construct the  $Z_{D2N}$  coarse space. We see that the iteration counts are almost constant as the size of the jump in the coefficients increases by six orders of magnitude.

In Figure 4.5, we plot the convergence curves for three successive refinements of the mesh. We see an increase in the number of iterations, which can be related to the

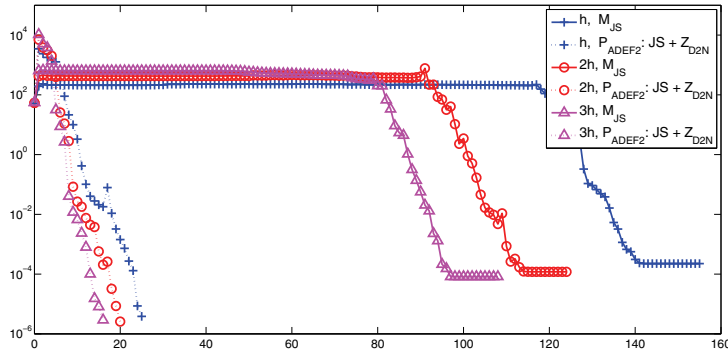


FIG. 4.6. HPL1 with various overlaps:  $h$  (crosses),  $2h$  (circles), and  $3h$  (triangles);  $n_{domain} = 32$ ,  $n_{xpt} = n_{ypt} = 16$ .  $Z_{D2N}$ : dotted lines, JS: solid lines.

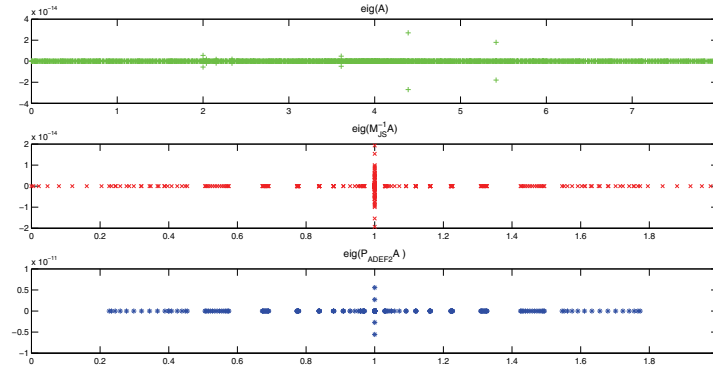


FIG. 4.7. Spectra of CONST case.  $n_{domain} = 16$ ,  $n_{overlap} = 1$ ,  $n_{xpt} = 8$ ,  $n_{ypt} = 16$ . The third row is the result of the  $Z_{D2N}$  method with one small eigenvector of the DtN map taken into account in each subdomain.

fact that the convergence rate of the Schwarz method depends on the physical size of the overlap. Here, we have a two element overlap, and thus the physical size of the overlap decreases as the mesh is refined.

In Figure 4.6, we consider the test HPL1 with various sizes of overlap,  $h$ ,  $2h$ , and  $3h$ , solved with the  $Z_{D2N}$  coarse space or without any coarse space (JS). For both methods, the iteration counts are improved as the overlap is increased. This is in agreement with the theory of the Schwarz method.

From the residual curves, we see that our method is very efficient and robust.

**4.1.3. Spectral analysis of the preconditioned system.** In Figures 4.7 and 4.8, we display the spectra of the CONST and HPL5 cases, respectively, for the original system, the one-level method, and the two-level method with the D2N coarse grid using  $P_{ADEF2}$ . The spectrum of the preconditioned matrices has three characteristics:

- the eigenvalues are between 0 and 2;
- the spectrum is more clustered;

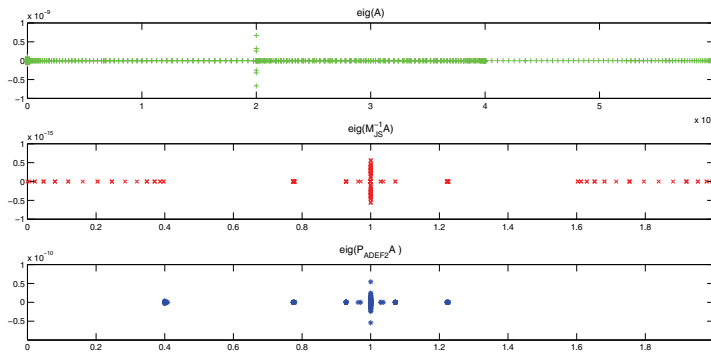


FIG. 4.8. Spectra of the HPL5 case.  $n_{\text{domain}} = 16$ ,  $n_{\text{overlap}} = 1$ ,  $n_{xpt} = 8$ ,  $n_{ypt} = 16$ . The third row is the result of the  $Z_{D2N}$  method with five small eigenvectors of the DtN map taken into account in each subdomain. The number of small eigenvalues is determined by (3.5).

TABLE 4.3

The minimum real part of the eigenvalues.  $\bar{A}$  is the coefficient matrix of the extended system (2.2), and  $M_{JS}$  is the JS preconditioner.  $n_{\text{domain}} = 16$ ,  $n_{\text{overlap}} = 1$ ,  $n_{xpt} = 8$ ,  $n_{ypt} = 16$ . The test cases are defined in Table 4.1.

Case	CONST	HPL1	HPL2	HPL3	HPL4	HPL5
$N_{smeig}$	1	1	2	3	4	5
$M_{JS}^{-1}\bar{A}$	2.67e-6	2.67e-6	2.67e-6	2.67e-6	2.67e-6	2.67e-6
$P_{ADEF2}\bar{A}$	0.23	0.40	0.40	0.40	0.40	0.40

- for  $Z_{D2N}$  the smallest eigenvalue is well separated from the origin.

Since the small eigenvalues near the origin have a negative influence on the fast convergence, we check the minimum real part of eigenvalues of six cases (see Table 4.3).

**4.2. General decompositions.** We now solve the model problem (1.1) on the domain  $\Omega = (0, 1)^2$  discretized by a  $P_1$  finite element method. The diffusion  $\kappa$  is a function of  $x$  and  $y$ . The BCs are zero-Dirichlet on the entire boundary. The corresponding discretizations and data structures were obtained by using the software FreeFem++ [21] in connection with a Metis partitioner [11]. In the following we will compare the AS and RAS preconditioners with and without Nicolaides coarse space to the new preconditioner based on the harmonic extension of the eigenvectors of the local DtN operators.

We test the method on overlapping decompositions into rectangular  $N \times N$  domains and on decompositions into  $N \times N$  irregular domains obtained via Metis. These overlapping decompositions are built by adding layers to nonoverlapping ones. The general nonoverlapping decompositions can be generated, for example, by using the Metis partitioner.

**4.2.1. Homogeneous viscosity.** In this case we have  $\kappa = 1$  in the whole domain. As in the case of a one-way decomposition, the two preconditioning methods based on Nicolaides coarse grid and DtN eigenvectors behave in quite a similar way whether AS or RAS are used when the viscosity is constant in the whole domain. Here we have chosen a  $4 \times 4$  regular subdomain decomposition. Each subdomain has 40 nodes on each side, and the overlap equals 2 elements. As expected, using the new algorithm, the number of eigenvectors contributed by each subdomain (which is

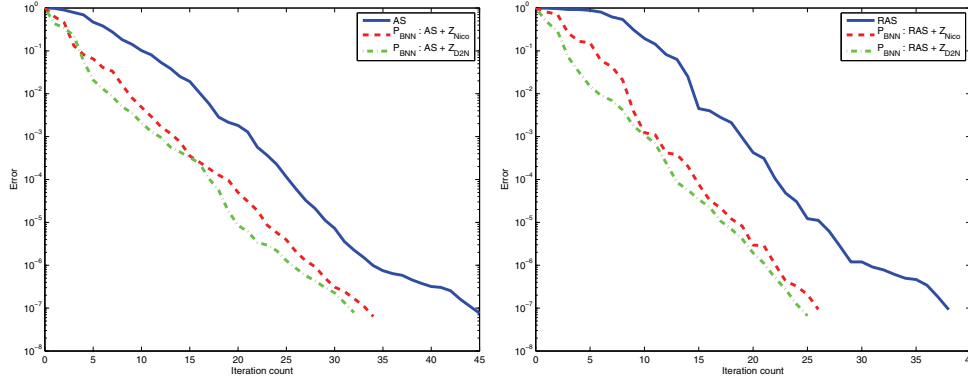


FIG. 4.9.  $4 \times 4$  subdomains, uniform  $\kappa$  distribution, for AS (left) and RAS (right).

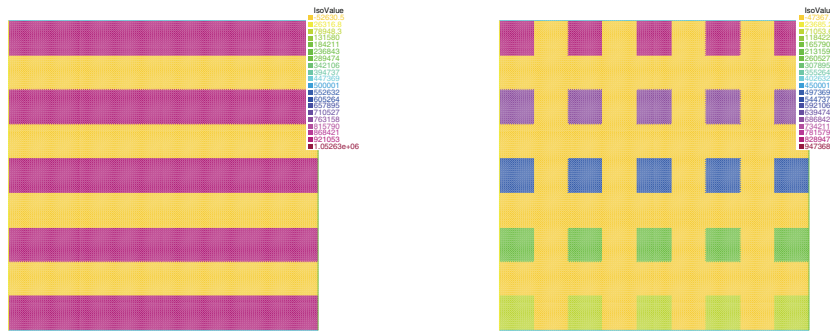


FIG. 4.10. Heterogeneous viscosity: alternating (left) and skyscraper (right) cases.

calculated automatically) is always 1. Figure 4.9 shows all three convergence curves for both AS and RAS. This numerically asserts that in the constant coefficient case the  $Z_{Nico}$  and  $Z_{D2N}$  are almost identical. The small differences lie in the fact that for  $Z_{Nico}$  the coarse space is defined analytically, whereas for  $Z_{D2N}$  it is obtained via a numerical procedure.

**4.2.2. Highly heterogeneous viscosity.** We will perform the same tests as before on two different configurations of highly heterogeneous viscosity. We consider the following situations:

- alternating viscosity: for  $y$  such that for  $[9y] \equiv 0 \pmod{2}$ ,  $\kappa = 10^6$ ; and  $\kappa = 1$  elsewhere. (See Figure 4.10(left).)
- skyscraper viscosity: for  $x$  and  $y$  such that for  $[9x] \equiv 0 \pmod{2}$  and  $[9y] \equiv 0 \pmod{2}$ ,  $\kappa = 10^5 \cdot ([9y] + 1)$ ; and  $\kappa = 1$  elsewhere. (See Figure 4.10(right).)

Here the number of eigenvectors considered in the construction of the coarse space is more important than in the homogeneous case (otherwise the global information cannot be correctly captured), and it may vary as a function of the number of subdomains. In the alternating case we used two or three modes per subdomain, whereas for the skyscraper case we used between two and four eigenvectors per subdomain. Note that for each subdomain the total number of eigenvalues is equal to the number of nodes on the interface. In these cases, typical values are 2, 3, or 4. Table 4.4 shows that using the new method highly improves convergence whether a uniform or

TABLE 4.4  
Convergence results for the skyscraper and alternating test cases.

	AS	AS+Nico	AS+D2N	RAS	RAS+Nico	RAS+D2N
Alternating	65	76	29	51	57	16
Metis alt.	101	90	37	83	67	23
Skyscraper	344	360	18	185	183	10
Metis sky.	375	968	28	164	158	19

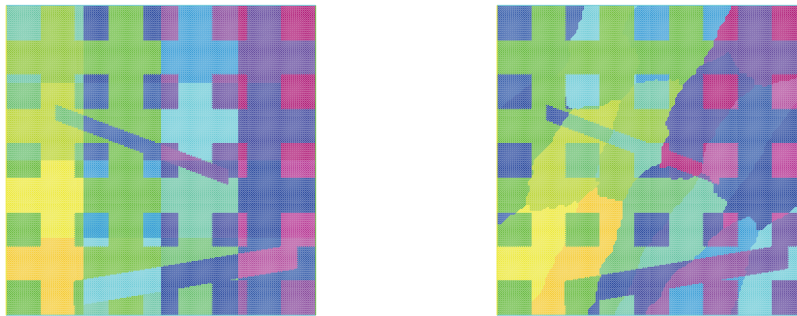


FIG. 4.11.  $4 \times 4$  subdomains uniform (left) and Metis (right). This shows both the subdomain boundaries and the jumps in  $\kappa$  for the “hard” test case.

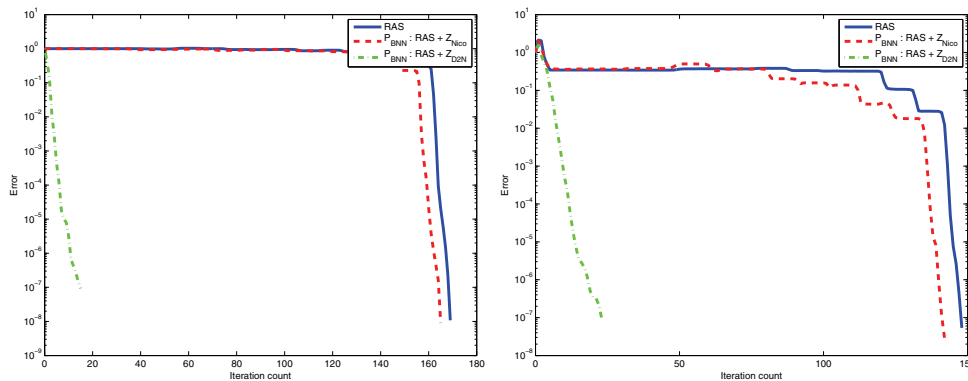


FIG. 4.12.  $4 \times 4$  subdomains, uniform (left) and Metis (right)—RAS convergence for the “hard” test case.

Metis partition is used in this highly heterogeneous coefficient case. The Nicolaidis preconditioner gives results comparable to those of the one-level method.

**4.2.3. A hard test case with inclusions and channels.** In order to compare our method to existing codes we solve a test case with known difficulties: the diffusion coefficient  $\kappa$  takes values between 1 and approximately  $1.5 \times 10^6$ , and the distribution contains both inclusions and channels. The total number of nodes will always be 256 on each side of  $\Omega$ . The decomposition will change, however: we will successively consider a  $2 \times 2$  decomposition, a  $4 \times 4$  decomposition, and an  $8 \times 8$  decomposition. *overlaps* will always be equal to 2. As an illustration, we have chosen to present the  $4 \times 4$  case extensively both with and without using the Metis partitioner.

Figure 4.11 shows together the decomposition into subdomains and the jumps in  $\kappa$ ; Figure 4.12 shows the convergence curves for the three methods using RAS. We



TABLE 4.5  
Convergence results for the hard test case.

	AS	AS+Nico	AS+D2N	RAS	RAS+Nico	RAS+D2N
$2 \times 2$	103	110	22	70	70	14
$2 \times 2$ with Metis	76	76	22	57	57	18
$4 \times 4$	603	722	26	169	165	15
$4 \times 4$ with Metis	483	425	36	148	142	23
$8 \times 8$	461	141	34	205	95	21
$8 \times 8$ with Metis	600	542	31	240	196	19

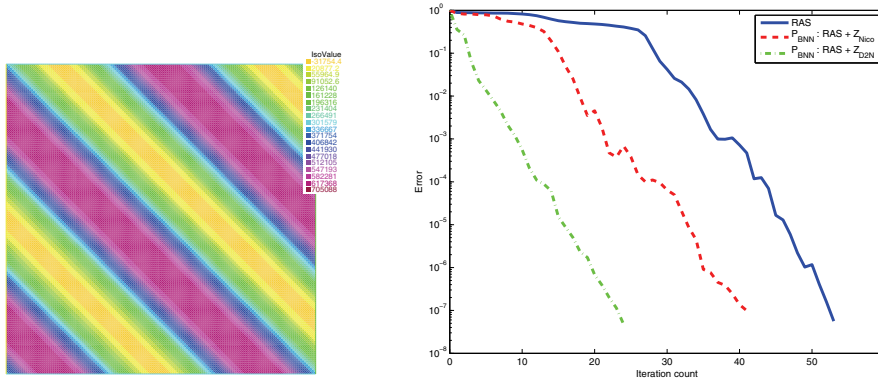


FIG. 4.13. Metis  $4 \times 4$  subdomains— $\kappa$  for the “continuous” test case (left) and convergence history (right).

TABLE 4.6  
Condition numbers for the continuous test case,  $8 \times 8$  decomposition with Metis.

	Condition number	Smallest eigenvalue	Largest eigenvalue
AS	4.566043e+03	8.760321e-04	4.000000e+00
AS + Nicolaides	1.127232e+02	3.547273e-02	3.998598e+00
AS + D2N	1.469540e+01	2.717523e-01	3.993508e+00

used between one and four eigenvectors per subdomain because the heterogeneities in  $\kappa$  impose that more global information on the solution be exchanged to avoid the plateau phenomenon. The convergence curves show that this strategy is a success; results are summarized in Table 4.5.

**4.2.4. Continuous variations of the coefficient.** This time we take an analytical function for  $\kappa$  which is chosen as

$$\kappa(x, y) = \kappa_M/3 \sin(\omega \pi (x + y) + 0.1).$$

In our case  $\kappa_M = 10^6$  and  $\omega = 4$ . We chose a two node overlap and a total of 160 nodes per side of the global domain. We study the  $4 \times 4$  and  $8 \times 8$  decompositions both with and without Metis. As an example, Figure 4.13 shows information for the  $4 \times 4$  decomposition using Metis. Table 4.6 shows information on the approximate condition numbers of the preconditioned operators in the Metis  $8 \times 8$  case. This was achieved using the Ritz values during the CG iteration procedure when AS is used. As expected for D2N, the smallest eigenvalue is larger than in the other cases, which leads to a better condition number for the operator. Finally, Table 4.7 shows the number of iterations needed to reach convergence for all three methods in all four

TABLE 4.7  
Convergence for the continuous test case.

	AS	AS+Nico	AS+D2N	RAS	RAS+Nico	RAS+D2N
$4 \times 4$	57	46	32	48	35	23
$4 \times 4$ with Metis	64	48	30	53	41	24
$8 \times 8$	461	141	34	205	95	21
$8 \times 8$ with Metis	600	542	31	240	196	19

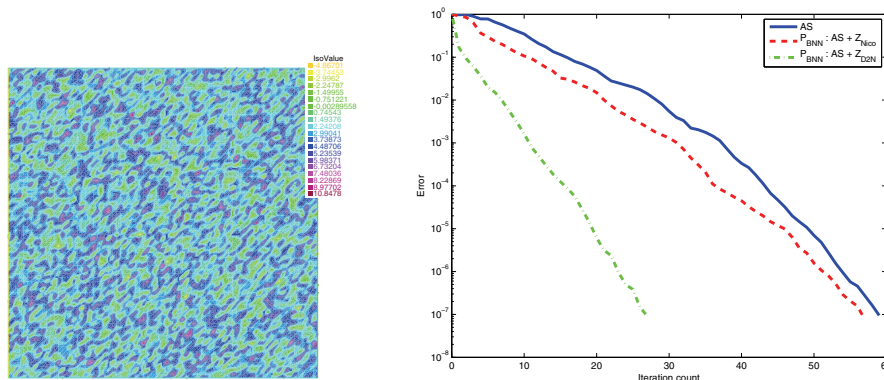


FIG. 4.14.  $\log(\kappa)$  for the random test case (left) and convergence history (right) for AS with an overlap of 2.

TABLE 4.8  
Condition numbers for the random test case and overlap = 2.

	Condition number	Smallest eigenvalue	Largest eigenvalue
AS	5.765809e+02	6.937447e-03	4.000000e+00
AS + Nicolaides	1.665329e+02	2.401928e-02	4.000000e+00
AS + D2N	1.635163e+01	2.445822e-01	3.999317e+00

cases. Again D2N performs significantly better.

In this section we propose a test with random coefficient. We use a log-normal distribution for the parameter  $\kappa$ . This distribution is calculated using the Gaussian random field generator available at [12]. We choose the exponential covariance family with parameters  $\theta_2 = 1$  and  $\theta_1 = e^{-\frac{1}{\lambda}}$  with  $\lambda = 4\Delta x$  in order for the correlation  $r(l)$  between the values at two points separated by a distance  $l$  to be  $r(l) = e^{-\frac{l}{\lambda}}$ . This gives us a normal distribution with mean value 0 of the random variable  $X$  at each grid point. We then define  $X' = \sigma X + \mu$  in order to have a  $\mu$  mean value and a  $\sigma$  standard deviation. In our example,  $\mu = 3$  and  $\sigma = 2$  (see Figure 4.14(left)). Finally, we calculate  $\kappa = 10^{X'}$ , giving us a log-normal random field with mean value  $\mu_\kappa = \mu \ln(10)$  and standard deviation  $\sigma_\kappa = \sigma \ln(10)$ .

**4.2.5. Random  $\kappa$  distribution.** The parameters of the test case are the following:  $4 \times 4$  subdomains, 20 nodes per subdomain edge, using a Metis partition. For these values  $\lambda = 4 * \frac{1}{4 \times 20} = 0.05$ . The size of the overlap will be successively 1, 2, and 3, and we will compare the RAS and AS methods. The size of the overlap does not have a significant influence on the comparison between methods, as shown in Table 4.9. In order to improve the condition number (Table 4.8) as many as 9 out of 99 eigenvectors are used to build the coarse space, with success.

TABLE 4.9  
*Convergence for the random test case ( $4 \times 4$  decomposition with Metis).*

	AS	AS+Nico	AS+D2N	RAS	RAS+Nico	RAS+D2N
Overlap = 1	89	82	30	79	78	22
Overlap = 2	59	57	27	50	52	33
Overlap = 3	45	46	22	36	40	24

**5. Conclusions.** We have considered the extended (2.2) and the original (2.1) linear systems arising from the domain decomposition method with overlap. We applied the two-level preconditioner using the Schwarz algorithm and the coarse-grid correction. The coarse-grid space is based on the low-frequency modes of the local DtN map. Its size automatically adapts to the difficulty of the problem. With this coarse space, we can obtain fast convergence for problems with large discontinuities (even along the interface) and arbitrary domain decompositions. The method has the potential to be extended to other systems of equations like elasticity, but it requires further investigation.

## REFERENCES

- [1] J. H. BRAMBLE, J. E. PASCIAK, AND A. H. SCHATZ, *The construction of preconditioners for elliptic problems by substructuring*, I, Math. Comput., 47 (1986), pp. 103–134.
- [2] X.-C. CAI AND M. SARKIS, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797.
- [3] A. ST-CYR, M. J. GANDER, AND S. J. THOMAS, *Optimized multiplicative, additive, and restricted additive Schwarz preconditioning*, SIAM J. Sci. Comput., 29 (2007), pp. 2402–2425.
- [4] C. R. DOHRMANN AND O. B. WIDLUND, *An overlapping Schwarz algorithm for almost incompressible elasticity*, SIAM J. Numer. Anal., 47 (2009), pp. 2897–2923.
- [5] C. R. DOHRMANN AND O. B. WIDLUND, *Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity*, Internat. J. Numer. Methods Engrg., 82 (2010), pp. 157–183.
- [6] M. DRYJA, M. V. SARKIS, AND O. B. WIDLUND, *Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions*, Numer. Math., 72 (1996), pp. 313–348.
- [7] E. EFSTATHIOU AND M. GANDER, *Why restricted additive Schwarz converges faster than additive Schwarz*, BIT, 43 (2003), pp. 945–959.
- [8] Y. A. ERLANGGA AND R. NABBEN, *Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 684–699.
- [9] Y. A. ERLANGGA AND R. NABBEN, *Algebraic multilevel Krylov methods*, SIAM J. Sci. Comput., 31 (2009), pp. 3417–3437.
- [10] J. F. ESCOBAR, *The geometry of the first non-zero Stekloff eigenvalue*, J. Funct. Anal., 150 (1997), pp. 544–556.
- [11] G. KARYPIS AND V. KUMAR, *METIS—Unstructured Graph Partitioning and Sparse Matrix Ordering System*; see <http://glaros.dtc.umn.edu/gkhome/views/metis>.
- [12] B. KOZINTSEV AND B. KEDEM, *Clipped Gaussian random fields*, field generating software, 1999, available online at <http://www-users.math.umd.edu/~bnk/bak/generate.cgi>.
- [13] F. MAGOULÈS, F.-X. ROUX, AND L. SERIES, *Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 3742–3759.
- [14] J. MANDEL, *Balancing domain decomposition*, Comm. Numer. Methods Engrg., 9 (1993), pp. 233–241.
- [15] J. MANDEL AND M. BREZINA, *Balancing domain decomposition for problems with large jumps in coefficients*, Math. Comp., 65 (1996), pp. 1387–1401.
- [16] F. NATAF, F. ROGIER, AND E. DE STURLER, *Optimal Interface Conditions for Domain Decomposition Methods*, Internal Report 301, CMAP (Ecole Polytechnique), Palaiseau, France, 1994; available online at <http://www.ann.jussieu.fr/~nataf/cvfiniMod.pdf>.

- [17] R. A. NICOLAIDES, *Deflation of conjugate gradients with applications to boundary value problems*, SIAM J. Numer. Anal., 24 (1987), pp. 355–365.
- [18] A. PADIY, O. AXELSSON, AND B. POLMAN, *Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 793–818.
- [19] C. PECHSTEIN AND R. SCHEICHL, *Scaling up through domain decomposition*, Appl. Anal., 88 (2009), pp. 1589–1608.
- [20] C. PECHSTEIN AND R. SCHEICHL, *Analysis of FETI methods for multiscale PDEs*, Numer. Math., 111 (2008), pp. 293–333.
- [21] O. PIRONEAU, F. HECHT, A. LE HYARIC, AND J. MORICE, *FreeFem++*, Laboratoire J.-L. Lions; available online at <http://www.freefem.org/ff++/>.
- [22] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
- [23] B. E. SMITH AND P. E. BJØSTAD, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.
- [24] J. M. TANG, R. NABBEN, C. VUIK, AND Y. A. ERLANGGA, *Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods*, J. Sci. Comput., 39 (2009), pp. 340–370.
- [25] J. M. TANG, S. P. MACLACHLAN, R. NABBEN, AND C. VUIK, *A comparison of two-level preconditioners based on multigrid and deflation*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1715–1739.
- [26] A. TOSELLI AND O. WIDLUND, *Domain Decomposition Methods: Algorithms and Theory*, Springer, New York, 2005.
- [27] C. VUIK, A. SEGAL, AND J. A. MEIJERINK, *An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients*, J. Comput. Phys., 152 (1999), pp. 385–403.
- [28] C. VUIK, A. SEGAL, J. A. MEIJERINK, AND G. T. WIJMA, *The construction of projection vectors for a deflated ICCG method applied to problems with extreme contrasts in the coefficients*, J. Comput. Phys., 172 (2001), pp. 426–450.