

# GRID TRANSFER OPERATORS FOR HIGHLY VARIABLE COEFFICIENT PROBLEMS IN TWO-LEVEL NON-OVERLAPPING DOMAIN DECOMPOSITION METHODS

L. GIRAUD\*, F. GUEVARA VASQUEZ \*, AND R. S. TUMINARO†

CERFACS Tech. Rep.: TR/PA/01/03

**Abstract.** We propose a robust interpolation scheme for non-overlapping two-level domain decomposition methods applied to two-dimensional elliptic problems with discontinuous coefficients. This interpolation is used to design a preconditioner closely related to the BPS scheme proposed in [1]. Through numerical experiments, we show on structured and unstructured finite element problems that the new preconditioning scheme reduces to the BPS method on smooth problems but outperforms it on problems with discontinuous coefficients. In particular it maintains good scalable convergence behavior even when the jumps in the coefficients are not aligned with subdomain interfaces.

**Keywords :** Domain decomposition, two-level preconditioning, Schur complement, parallel distributed computing, elliptic partial differential equations, discontinuous coefficients.

**1. Introduction.** In recent years, there has been significant work on domain decomposition algorithms for numerically solving partial differential equations. Several numerically scalable domain decomposition preconditioners possess optimal convergence rates when used with Krylov methods for given classes of elliptic problems. These optimality or quasi-optimality properties are achieved with two-level preconditioners that use both local and global approximations either in an additive or multiplicative way. The first two-level preconditioner (BPS) for non-overlapping domain decomposition techniques was introduced by Bramble, Pasciak and Schatz [1]. In their paper, the authors showed that for a uniformly elliptic operator the condition number of the preconditioned system does not depend on the number of subdomains and only weakly depends on the number of mesh points within subdomains. Other non-overlapping domain decomposition preconditioners that possess similar optimality properties include the vertex space [5, 14], the balancing Neumann-Neumann [9, 10, 11] and the FETI [6, 12] methods. For most of these techniques, this property can also be extended to discontinuous coefficient problems under the assumption that the jumps in the coefficients align with the interfaces between subdomains [8, 9, 15, 19]. While domain decomposition techniques can be applied to situations where interfaces are not aligned, their performance is generally less good than for constant coefficient problems. In this paper, we present a new interpolation operator that is intended to address discontinuous coefficients where interfaces are not aligned with subdomains. The new interpolation operator was first introduced in [7] and some preliminary results on structured meshes were presented in [3]. To illustrate the benefits of the new grid transfer method, it is used to modify a specific two-level BPS-like method. The original preconditioner (referred to as BPS\*) presented in [4] is intended to be representative of other BPS-like methods. For smooth problems the new preconditioner reduces to regular BPS\* but it performs much better on problems where the jumps do not coincide with the boundary of the subdomains. The robust-

---

\*CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse Cedex, France.

†Sandia National Laboratory, Livermore, CA, 94551 USA. Supported by the Applied Mathematical Sciences program, U.S. Department of Energy, Office of Energy Research, and was performed at Sandia National Laboratories, operated for the U.S. Department of Energy under contract No. DE-AC04-94AL85000.

ness of the new preconditioner is assessed through extensive numerical experiments both on structured and unstructured meshes.

The paper is organized as follows. In Section 2 we briefly present the BPS\* preconditioner and describe the interpolation operator in the framework of structured meshes. In Section 3 we propose a generalization of the interpolation for unstructured meshes. In both situations we report numerical experiments to illustrate the attractive behavior and the robustness of the new preconditioner compared with the BPS\* method.

**2. Structured meshes.** We consider the following  $2^{nd}$  order self-adjoint elliptic problem on an open polygonal domain  $\Omega$  included in  $\mathbb{R}^2$ :

$$\begin{cases} -\frac{\partial}{\partial x}(a(x,y)\frac{\partial v}{\partial x}) - \frac{\partial}{\partial y}(b(x,y)\frac{\partial v}{\partial y}) = F(x,y) & \text{in } \Omega, \\ v = 0 & \text{on } \partial\Omega, \end{cases} \quad (2.1)$$

where  $a(x,y), b(x,y) \in \mathbb{R}^2$  are bounded positive functions on  $\Omega$ . We discretize (2.1) via finite elements resulting in a sparse symmetric and positive definite (possibly unstructured) matrix equation

$$Au = f.$$

We assume that the domain  $\Omega$  is partitioned into  $N$  non-overlapping subdomains  $\Omega_1, \dots, \Omega_N$  with boundaries  $\partial\Omega_1, \dots, \partial\Omega_N$ . Let  $B$  be the set of all indices of the discretized points which belong to the interfaces between the subdomains and  $I$  be the set of all indices which correspond to subdomain interiors. Grouping the points corresponding to  $B$  in the vector  $u_B$  and those corresponding to  $I$  in the vector  $u_I$  induces the reordered problem:

$$\begin{pmatrix} A_{II} & A_{IB} \\ A_{IB}^T & A_{BB} \end{pmatrix} \begin{pmatrix} u_I \\ u_B \end{pmatrix} = \begin{pmatrix} f_I \\ f_B \end{pmatrix}. \quad (2.2)$$

Eliminating  $u_I$  from the second block row of (2.2) leads to the following reduced equation for  $u_B$ :

$$Su_B = f_B - A_{IB}^T A_{II}^{-1} f_I, \quad \text{where} \quad S = A_{BB} - A_{IB}^T A_{II}^{-1} A_{IB} \quad (2.3)$$

is the Schur complement of the matrix  $A_{II}$  in  $A$ , and is usually referred to as the Schur complement matrix. The reduced system (2.3) is usually solved via a preconditioned conjugate gradient (PCG) method as the Schur complement matrix inherits the symmetric positive definiteness property from  $A$ .

To describe the Schur complement preconditioners, we need to define a partition of  $B$ . Let  $V_i$  be the singleton sets that contain one index related to one cross point (where two or more edges meet) and let  $V = \cup_i V_i$  be the set with all those indices; each cross point is represented by an  $\times$  in Figure 2.1. We define each edge by considering neighboring subdomains  $\Omega_j$  and  $\Omega_l$  for  $j \neq l$ ,  $(j, l) \in \{1, 2, \dots, N\}^2$  where subdomain  $j$  and  $l$  are neighboring if  $\partial\Omega_j \cap \partial\Omega_l$  is nonempty and contains at least one point not in  $V$ . We define each edge  $E_i$  by

$$E_i = (\partial\Omega_j \cap \partial\Omega_l) \setminus V.$$

In Figure 2.1, the points belonging to the  $m$  edges  $(E_i)_{i=1,m}$  are represented by  $\bullet$ . We can thus describe the set  $B$  as

$$B = \left( \bigcup_{i=1}^m E_i \right) \cup V, \quad (2.4)$$

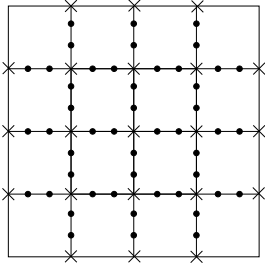


FIG. 2.1. A  $4 \times 4$  box-decomposition with edge ( $\bullet$ ) and vertex ( $\times$ ) points.

consisting of  $m$  edges and the cross points.

We now consider BPS-like preconditioners [1]. In this paper, we choose the BPS\* method [4]. This scheme is closely related to the classical BPS algorithm and so serves as a good representative of BPS-like methods. It is important to understand that the operator dependent interpolation discussed in this paper is not limited to this specific BPS\* method and is applicable to any BPS variant. The BPS\* preconditioner can be briefly described as follows. We first define a series of projection and interpolation operators. Specifically, for each  $E_i$  we define an  $|E_i| \times |B|$  matrix<sup>1</sup>,  $R_i = R_{E_i}$ , as the standard point-wise restriction (or injection) of nodal values to  $E_i$ . Its transpose extends grid functions in  $E_i$  by zero on the rest of the interface  $B$ . Similarly we define the  $|V| \times |B|$  matrix,  $R_V$ , as the canonical restriction to  $V$ . Thus, we set  $S_{ij} \equiv R_i S R_j^T$  and  $S_V \equiv R_V S R_V^T$ . To complete the two-level preconditioner, a coarse grid operator must be defined. Assume that  $\Omega_1, \dots, \Omega_N$  form the elements of a coarse grid mesh,  $\tau^H$ , with mesh size  $H$ . That is, a coarse grid mesh point,  $\hat{V}_j$ , is associated with each fine grid vertex point,  $V_j$ , and two coarse grid points,  $\hat{V}_j$  and  $\hat{V}_k$ , are adjacent if and only if there is an  $E_i$  that connects their corresponding fine grid points,  $V_j$  and  $V_k$ . An interpolation operator is defined by injecting coarse grid values to the corresponding fine grid cross points and performing linear interpolation for each adjacent pair of coarse grid points along the edge connecting them. This linear interpolation operator is denoted  $R_l^T$ .  $R_l$  is a projection operator and is the transpose of the interpolation operator. Finally,  $A_{H,l}$  is the Galerkin coarse grid operator

$$A_{H,l} = R_l A R_l^T \quad (2.5)$$

defined on  $\tau^H$ . With these notations the BPS\* preconditioner is defined by

$$M_{BPS^*} = \sum_{E_i} R_i^T S_{ii}^{-1} R_i + R_l^T A_{H,l}^{-1} R_l, \quad (2.6)$$

as described in [4]. Within this paper we consider the use of  $S_{ii}^{-1}$  to simplify the comparisons between methods. However, it should be noted that within practical computations,  $S_{ii}^{-1}$  is typically replaced with an inexpensive spectrally equivalent approximation (e.g. [16]). This preconditioner (2.6) can be interpreted as a generalized block Jacobi scheme for the Schur complement system (2.3) where the block diagonal preconditioning for  $S_{VV}$  is omitted and a coarse grid correction is added. The coarse grid term  $R_l^T A_{H,l}^{-1} R_l$  incorporates global coupling between distant interfaces.

<sup>1</sup> $|\cdot|$  denotes cardinality.

This global coupling is critical for scalability. In particular, it has been shown in [1] that when applying the original BPS technique to a uniformly elliptic operator, the preconditioned system has a condition number

$$\kappa(M_{BPS}S) = \mathcal{O}(1 + \log^2(H/h)) \quad (2.7)$$

where  $h$  is the mesh size. This result can be extended to the situation of discontinuous coefficients under the assumption that the jumps occur at interfaces between subdomains [1].

In this paper, we replace  $A_{H,l}$  by

$$\tilde{A}_{H,l} = R_l S R_l^T. \quad (2.8)$$

It has been shown [1, 16] that in general for elliptic problems the operator  $A_{H,l}$  is spectrally equivalent to  $\tilde{A}_{H,l}$  (in fact they are equal in some situations) and so it is possible to use either  $A_{H,l}$  or  $\tilde{A}_{H,l}$ . In this text, we address only problems for which this spectral equivalence between (2.5) and (2.8) is ensured. We generally find  $\tilde{A}_{H,l}$  more robust when highly discontinuous coefficients are present and so we use (2.8) for our convergence experiments though within practical computations  $A_{H,l}$  may be more attractive. Thus, a linear interpolation modified BPS\* scheme is given by

$$\tilde{M}_{BPS^*,l} = \sum_{E_i} R_i^T S_{ii}^{-1} R_i + R_l^T \tilde{A}_{H,l}^{-1} R_l. \quad (2.9)$$

This method will be compared to a preconditioner using operator dependent interpolation. Specifically, a specialized interpolation transfer (operator dependent) that addresses jumps that are not aligned with subdomain boundaries is defined in the next section. This operator is denoted  $R_{od}$  and replaces  $R_l$  yielding an operator dependent version of a BPS\* scheme

$$\tilde{M}_{BPS^*,od} = \sum_{E_i} R_i^T S_{ii}^{-1} R_i + R_{od}^T \tilde{A}_{H,od}^{-1} R_{od} \quad (2.10)$$

where

$$\tilde{A}_{H,od} = R_{od} S R_{od}^T.$$

**2.1. Operators Dependent Interpolation.** When uniform rectangular subdomains are used on a two-dimensional structured mesh, all the edges  $E_i$  are aligned either with the  $x$  or the  $y$  axis. In this case, interpolation,  $R_{od}^T$ , reduces to a series of one-dimensional interpolations along either  $x$  or  $y$  grid lines. Therefore, we begin by considering one-dimensional interpolation. Clearly grid points on each straight-line that contain edges can be mapped to points on a line on the interval  $(0,1)$ . We, therefore, consider the one dimensional model problem

$$\begin{cases} -\frac{d}{dx}(a(x)\frac{d}{dx}u(x)) = f & \text{in } (0,1), \\ u(x) = 0 & \text{at } x=0 \text{ and } 1. \end{cases} \quad (2.11)$$

In the discussion that follows, we will assume that  $A$  is discretized via linear finite elements in order to simplify the presentation. Let  $H^1(0,1)$  be the standard Sobolev

space on the interval  $(0,1)$  and  $H_0^1(0,1)$  its subspace whose functions vanish at  $x = 0$  and  $x = 1$ . Given a grid  $x_j^h = jh, j = 0, \dots, n$  on  $(0,1)$ , define the fine grid linear finite element space to be

$$V^h = \{v^h \in H_0^1(0,1) : v^h \text{ is linear on } [x_j^h, x_{j+1}^h], j = 0, \dots, n-1\},$$

and denote the set of nodal basis functions by  $\{\phi_j^h\}_{j=1}^{n-1} \in V^h$  where

$$\phi_j^h(x_k^h) = \begin{cases} 1 & \text{if } k = j, \\ 0 & \text{if } k \neq j. \end{cases}$$

Note that the  $\phi_j^h$ 's span  $V^h$ . Let  $(x_i^H)_{i=1,m}$  be the set of coarse grid points defined by the interfaces generated when partitioning  $(0,1)$  into  $m+1$  non-overlapping subdomains. Define the coarse subspace  $V^H = \text{span}\{\phi_i^H \in V^h : i = 1, \dots, m\}$  where the coarse grid nodal basis functions satisfy

$$\phi_j^H(x_k^H) = \begin{cases} 1 & \text{if } k = j, \\ 0 & \text{if } k \neq j. \end{cases}$$

Note that the  $\phi_i^H$ 's are linear on  $[x_j^h, x_{j+1}^h]$  and are not yet fully specified. Since the  $\{\phi_i^h\}$  are linearly independent and the  $\{\phi_i^H\}$  lie in  $V^h$ , there exists a unique matrix  $R$  of size  $m \times (n-1)$  such that

$$[\phi_1^H \dots \phi_m^H] = [\phi_1^h \dots \phi_{n-1}^h] R^T.$$

Thus, defining the  $\phi_i^H$ 's is equivalent to building a restriction operator,  $R$ , and a corresponding interpolation operator,  $R^T$ .

We depict in Figure 2.2 an example of a coarse grid basis function that defines linear interpolation. If we denote  $\tilde{u} \in V_h$  as the linear interpolant of the function

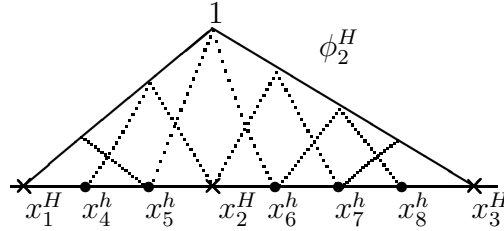


FIG. 2.2. coarse grid basis function  $\phi_2^H$  associated with the linear interpolation.

$u \in V_H$ , the product  $a \frac{\partial \tilde{u}}{\partial x}$  is discontinuous at points where  $a(\cdot)$  is discontinuous. Intuitively, the discontinuity of  $a \frac{\partial \tilde{u}}{\partial x}$  is undesirable as this term is further differentiated in (2.11). The undesirability of a discontinuous  $a \frac{\partial \tilde{u}}{\partial x}$  can also be motivated using energy minimization arguments (normally used in algebraic multigrid discussions [13]). In particular, the coarse grid system should correct smooth error components,  $e$ . These smooth components are typically characterized by small energy. That is,  $\langle e, Ae \rangle \ll \langle e, e \rangle$  where  $A$  is scaled so  $\|A\| = 1$  and  $\langle \cdot, \cdot \rangle$  defines the usual vector inner product. However, the discontinuity of  $a \frac{\partial \tilde{u}}{\partial x}$  implies that a high energy error correction will be applied. That is,  $\langle \tilde{u}, A\tilde{u} \rangle$  is large where  $\tilde{u}$  is the interpolated coarse grid correction. This is clearly contrary to what is needed and so the interpolation must be remedied



LEMMA 1. *The operator-dependent interpolation reduces to linear interpolation when  $a \equiv 1$ .*

**Proof**

When  $a() \equiv 1$ , the solution to (2.12) such that  $\phi_i^H(x_{i-1}^H) = 0$  and  $\phi_i^H(x_i^H) = 1$  is

$$\phi_i^H = \frac{x - x_{i-1}^H}{x_i^H - x_{i-1}^H}$$

which is the straight line that defines linear interpolation as depicted in Figure 2.2. ■

LEMMA 2. *The one dimensional operator-dependent interpolation defined by the solution of (2.12) reduces to the multigrid energy minimization approach [18] or to the multigrid operator-dependent interpolation when every other point is a coarse point (i.e. each subdomain contains only one point).*

LEMMA 3. *This new interpolation defines a partition of unity. That is,*

$$R_{od}^T \mathbf{1}_V = \mathbf{1}_B, \quad (2.14)$$

where the symbol  $\mathbf{1}_V$  denotes the vectors of all 1's on the coarse grid and  $\mathbf{1}_B$  is the vector of 1's on all the subdomain edges and cross points.

**Proof**

It is enough to show that  $\sum_i \phi_i^H(x) = 1$ . That is, let  $\psi^H(x) = \sum_i \phi_i^H(x)$ . Using (2.12) with  $1 \leq i \leq m$ , it follows that

$$\begin{cases} -\frac{d}{dx}(a(x)\frac{d}{dx}\psi_i^H) = 0 & \text{in } [x_{i-1}^H, x_i^H] \\ \psi_i^H(x_{i-1}^H) = 1, \quad \psi_i^H(x_i^H) = 1. \end{cases}$$

Clearly,  $\psi^H = 1$  is a solution and by uniqueness,  $\psi^H \equiv 1$  on  $[x_{i-1}^H, x_i^H]$  and the result follows. ■

It should be noted that this proof is essentially identical to the one given in [18, p. 1637] for Lemma 3.2 with the exception that here the number of fine grid points within  $[x_{i-1}^H, x_i^H]$  is arbitrary. This partition of unity is generally critical for good numerical convergence and occurs in most multilevel methods: the Neumann-Neumann and balancing Neumann-Neumann preconditioner [9, 10], etc. Connections between this grid transfer and standard operator dependent multigrid transfers can be found in [7].

**2.2. Numerical experiments.** To evaluate the sensitivity of the preconditioners to discontinuity, we consider (2.1) where the coefficients  $a()$  and  $b()$  are defined in several different ways. In all of our examples, these diffusion coefficients are piecewise constant functions. For the first two problems  $a()$  and  $b()$  are given by

$$\begin{aligned} \text{- Problem SD-F1:} \quad a() = b() &= \begin{cases} 1 & \text{in } \mathcal{R}^3, \\ 10^2 & \text{in } \mathcal{R}^2 \cup \mathcal{R}^4, \\ 10^{-2} & \text{in } \mathcal{R}^1 \cup \mathcal{R}^5. \end{cases} \\ \text{- Problem SD-F2:} \quad a() = b() &= \begin{cases} 1 & \text{in } \mathcal{R}^3, \\ 10^3 & \text{in } \mathcal{R}^2 \cup \mathcal{R}^4, \\ 10^{-3} & \text{in } \mathcal{R}^1 \cup \mathcal{R}^5. \end{cases} \end{aligned}$$

where the  $\mathcal{R}$ 's are given by Figure 2.4. The third problem uses

- Problem SD-R: 
$$a() = b() = \begin{cases} 10^{-1} & \text{in } \mathcal{R}^1, \\ 10^{-2} & \text{in } \mathcal{R}^2, \\ 10^1 & \text{in } \mathcal{R}^3 \cup \mathcal{R}^4 \cup \mathcal{R}^5 \cup \mathcal{R}^6. \end{cases}$$

where the  $\mathcal{R}$ 's are given by Figure 2.5. For sake of completeness we also consider the classical Poisson problem defined by  $a() = b() = 1$ .

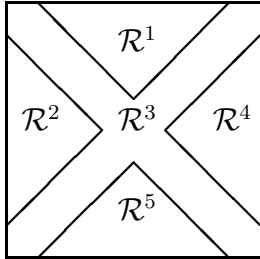


Figure 2.4: Example 1 - Flag

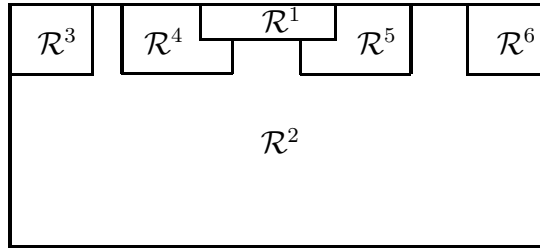


Figure 2.5: Example 2 - Region

For all these experimental results, structured meshes are used along with uniform square subdomains. The convergence of the preconditioned conjugate gradient method is attained when the 2-norm of the residual normalized by the 2-norm of the right hand side is less than  $10^{-8}$ . The experiments were run on the ASCI Red machine. In all these simulations, each subdomain is assigned to a different processor and the coarse grid problem is replicated on each processor. More details on the parallel implementation can be found in [17].

We study the numerical scalability of the preconditioners by investigating the dependence of the convergence on the number of the subdomains while keeping constant the number of grid points per subdomain (i.e.  $256 \times 256$  mesh for each subdomain, that is  $\frac{H}{h} = 256$ ). The initial guess  $x_0$  for the conjugate gradient iterations is the null vector and the righthand side is given by a constant vector. In Table 2.1, we give the results for the Poisson problem. As expected, the behavior of the two variants does not depend on the number of subdomains as predicted by (2.7) and as indicated by Lemma 1. In Table 2.2 we report the number of iterations on heterogeneous problems. Notice that for the test examples the discontinuities in the coefficients  $a()$  and  $b()$  are not aligned with the interface of the square subdomains and consequently no theory applies. These results reveal that  $\tilde{M}_{BPS^*,l}$  is less efficient than  $\tilde{M}_{BPS^*,od}$ . Further,  $\tilde{M}_{BPS^*,od}$  is numerically scalable (as is  $\tilde{M}_{BPS^*,l}$ ) and follows a behavior predicted by (2.7) even though this theory does not apply.

### 3. Generalization to unstructured meshes.

**3.1. Definition of the interpolation operator.** For unstructured meshes, the interfaces between subdomains are not necessarily aligned with either the  $x$  or  $y$  axis. Therefore, the one-dimensional PDE solution procedure involving either the coefficient function  $a()$  or  $b()$  to define (2.12) cannot be used anymore. However, we can still construct a linear system similar to (2.13) to build an interpolation operator. In particular for each interface, we consider all the elements containing an edge along the interface. For each element, we take its stiffness matrix and algebraically eliminate (i.e. reduce out) all unknowns that do not lie on the interface. Then, for each interface a matrix is assembled that includes contributions from all its reduced stiffness matrices. Finally, the two matrix rows corresponding to the end points (i.e. the two coarse grid points) are replaced by Dirichlet boundary conditions. It is important to notice that

# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$	$32 \times 32$
$M_{BPS*,l}$	13	17	17	15
$\tilde{M}_{BPS*,od}$	13	17	17	15

TABLE 2.1

# iterations to solve the Poisson problem on structured meshes.

SD-F1 problem				
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$	$32 \times 32$
$M_{BPS*,l}$	26	33	32	32
$\tilde{M}_{BPS*,od}$	22	22	27	23
SD-F2 problem				
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$	$32 \times 32$
$M_{BPS*,l}$	28	36	36	38
$\tilde{M}_{BPS*,od}$	23	24	31	26
SD-R problem				
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$	$32 \times 32$
$M_{BPS*,l}$	18	35	37	37
$\tilde{M}_{BPS*,od}$	17	24	22	19

TABLE 2.2

# iterations to solve PDEs with discontinuities on structured meshes.

by reducing out the noninterface unknowns, a tridiagonal matrix is obtained. We depict in Figure 3.1 an example interface for an unstructured mesh. The shaded elements are those used to construct the tridiagonal system. For the interpolant corresponding to the left coarse grid point, the right hand side contains a ‘one’ at the left Dirichlet condition, a ‘zero’ at the right Dirichlet condition and ‘zeros’ for all other equations. Notice that this linear system effectively coincides to the original PDE on a strange domain with Neumann boundary conditions everywhere except at the two end points where Dirichlet boundary conditions are enforced. This use of local stiffness matrices has been found beneficial in other multilevel methods such as the AMGe algebraic multigrid algorithm [2], the FETI method [6], and the balancing Neumann-Neumann scheme [10]. In all of these algorithms, the local stiffness matrices provides a local submatrix which is closely connected to the original physical problem (e.g. a substructure).

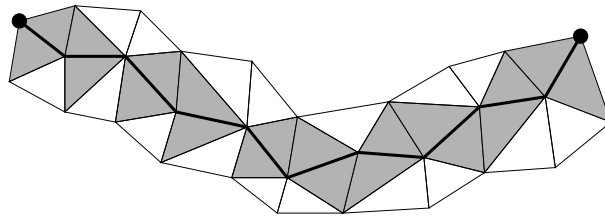


FIG. 3.1. An interface on unstructured meshes.

LEMMA 4. *The operator-dependent interpolation constructed on unstructured meshes defines a partition of the unity.*

**Proof**

Let  $A^{(s)}$  be the elementary  $k \times k$  stiffness matrix associated with element  $s$ . This matrix has the form

$$A^{(s)} = \begin{pmatrix} A_{\ell\ell} & A_{\ell e} \\ A_{e\ell} & A_{ee} \end{pmatrix}$$

where the first block of rows are associated with noninterface nodes and the second block of rows correspond to the interface. For any element,  $s$ , the rank of  $A^{(s)}$  is equal to  $k - 1$  and its null space is spanned by the constant vector (i.e.  $A^{(s)} \mathbf{1}_k = \mathbf{0}_k$ ). This implies that the local Schur complement

$$A_{ee} - A_{e\ell} A_{\ell\ell}^{-1} A_{\ell e}$$

is also rank deficient with a null space still spanned by the constant vector. Clearly, this continues to hold when assembling the reduced stiffness matrices. That is, any constant vector is a solution to the assembled reduced system with a zero righthand side and no Dirichlet boundary conditions. Imposing Dirichlet boundary conditions equal to one at the endpoints (i.e. interpolating the coarse grid vector  $\mathbf{1}_V$ ) forces the unique solution to be the vector of all ones along the interface. ■

REMARK 1. *On structured meshes decomposed into uniform rectangles, the above definition leads to the same operator-dependent interpolation defined in the previous section for structured meshes.*

While our algorithm eliminates noninterface nodes at the elementary level (i.e. before assembling), this elimination could be done after assembling. This would correspond to defining interpolation as the solution of

$$\left\{ \begin{array}{ll} -\frac{\partial}{\partial x}(a(x, y) \frac{\partial v}{\partial x}) - \frac{\partial}{\partial y}(b(x, y) \frac{\partial v}{\partial y}) = 0 & \text{on } \hat{\Omega}, \\ \frac{\partial v}{\partial \mathbf{n}} = 0 & \text{on } \partial \hat{\Omega} \setminus \{V_i, V_j\}, \\ v = 1 & \text{at } \{V_i\} \\ v = 0 & \text{at } \{V_j\} \end{array} \right. \quad (3.1)$$

where  $\hat{\Omega}$  is the union of the elements that have an edge along the interface and  $\{V_i, V_j\}$  refers to the two cross points corresponding to the edge endpoints (see Figure 3.1). This possibility has not been explored. Our experiences with averaging on structured meshes lead us to believe that possible benefits do not outweigh the additional costs.

**3.2. Numerical experiments.** To investigate the robustness and the scalability of the preconditioners we consider three model problems by defining the diffusion coefficients  $a()$  and  $b()$  in (2.1) as piecewise constant functions in  $(-1, 1)^2$  as depicted in Figures 3.2, 3.3 and 3.4. Using these notations we define a set of model problems with discontinuous coefficients as follows:

$$\text{- Problem UD-B: } a() = b() = \begin{cases} 10^{-2} & \text{in } \mathcal{R}^1 \cup \mathcal{R}^7, \\ 10^2 & \text{in } \mathcal{R}^2 \cup \mathcal{R}^6, \\ 10^{-3} & \text{in } \mathcal{R}^3 \cup \mathcal{R}^5, \\ 10^3 & \text{in } \mathcal{R}^4; \end{cases}$$

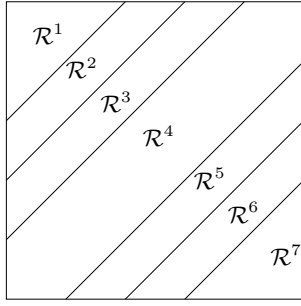


FIG. 3.2. *Example 3 - Band.*

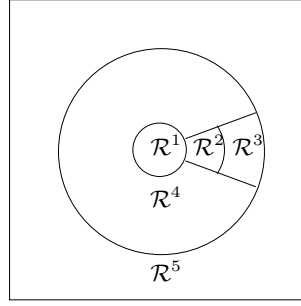


FIG. 3.3. *Example 4 - Radar.*

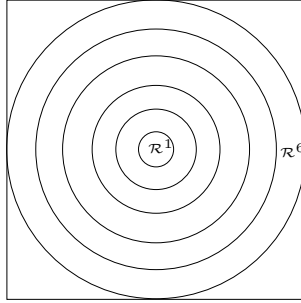


FIG. 3.4. *Example 5 - Ring.*

where the  $\mathcal{R}^i$  are defined in Figure 3.2,

$$\text{- Problem UD-Ra: } a() = b() = \begin{cases} 10^{-3} & \text{in } \mathcal{R}^1, \\ 10^2 & \text{in } \mathcal{R}^2, \\ 10 & \text{in } \mathcal{R}^3, \\ 10^{-2} & \text{in } \mathcal{R}^4, \\ 10^{-1} & \text{elsewhere;} \end{cases}$$

where the  $\mathcal{R}^i$  are defined in Figure 3.3,

$$\text{- Problem UD-Ri: } a() = b() = \begin{cases} 10^3 & \text{in } \mathcal{R}^1, \\ 10^2 & \text{in } \mathcal{R}^2, \\ 10^1 & \text{in } \mathcal{R}^3, \\ 10^{-3} & \text{in } \mathcal{R}^4, \\ 10^{-1} & \text{in } \mathcal{R}^5, \\ 1 & \text{in } \mathcal{R}^6, \\ 10^{-1} & \text{elsewhere;} \end{cases}$$

where the  $\mathcal{R}^i$  are defined in Figure 3.4. The rings are labeled consecutively starting with the innermost ring labeled as  $\mathcal{R}^1$ .

We consider the following geometries and unstructured meshes to study the scalability of the preconditioners:

- rsq** Each subdomain is a rounded square as depicted in Figure 3.5 (i.e. a square where the sides have been replaced by arcs except on the domain boundary).

- par** Each subdomain is a parallelogram (see Figure 3.6).
- rpar** Each subdomain is a “rounded parallelogram” as depicted in Figure 3.7 (i.e. a parallelogram where the sides have been replaced by arcs except on the domain boundary).

Note that all the meshes preserve the external domain geometry. That is, the entire

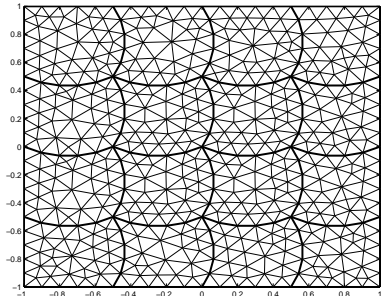


FIG. 3.5. *rsq unstructured mesh: a  $4 \times 4$  decomposition.*

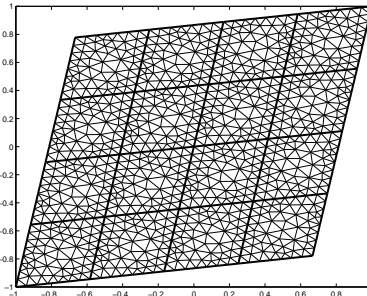


FIG. 3.6. *par unstructured mesh: a  $4 \times 4$  decomposition.*

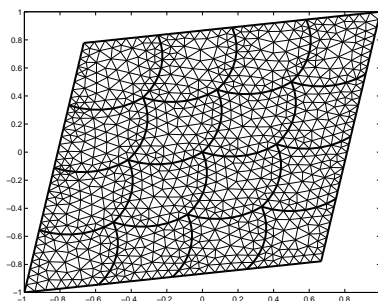


FIG. 3.7. *rpar unstructured mesh: a  $4 \times 4$  decomposition.*

composite domain  $\Omega$  remains the same when the number of subdomains changes.

The mesh generation and the finite element discretization of (2.1) are done using MATLAB’s PDE toolbox. In our implementation, building  $\tilde{M}_{BPS^*,od}$  costs a little bit more than constructing  $\tilde{M}_{BPS^*,l}$  due to a couple of additional nearest neighbor communications. The computational cost for applying both methods, however, is identical. Solutions are obtained in parallel on the ASCI Red machine. Even though on unstructured meshes it is more difficult to control the ratio  $\frac{H}{h}$ , we attempt to keep it constant as well as maintain a good aspect ratio for the subdomains. Therefore, we can still study the scalability of the preconditioners varying the number of subdomains while the ratio  $\frac{H}{h}$  remains the same. Table 3.1 gives the range of number of points per subdomain for each mesh.

We first report in Table 3.2 the numerical behavior of the two preconditioners for the Poisson problem. It can be observed that for that problem, the two preconditioners have similar convergence behavior. In addition the convergence is independent of the

Domain Decomposition	approx. range of pts/sd
Parallelograms ( <b>par</b> )	890 - 900
Round Squares ( <b>rsq</b> )	710 - 860
Round Parallelograms ( <b>rpar</b> )	780 - 1100

TABLE 3.1

*Approximate range of points per subdomain for the considered domain decompositions.*

Round square mesh (rsq)			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\tilde{M}_{BPS^*,l}$	16	17	16
$\tilde{M}_{BPS^*,od}$	18	18	17
Parallelogram mesh (par)			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\tilde{M}_{BPS^*,l}$	14	14	13
$\tilde{M}_{BPS^*,od}$	14	14	13
Round parallelogram mesh (rpar)			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\tilde{M}_{BPS^*,l}$	20	19	18
$\tilde{M}_{BPS^*,od}$	21	21	20

TABLE 3.2

*# iterations to solve the Poisson problem on unstructured meshes.*

number of subdomains as predicted by (2.7). In Tables 3.3-3.5 we report the number of PCG iterations for  $\tilde{M}_{BPS^*,l}$  and  $\tilde{M}_{BPS^*,od}$  on discontinuous problems discretized on unstructured meshes. For each mesh, we vary the number of subdomains from 16 to 256. As it can be observed,  $\tilde{M}_{BPS^*,od}$  performs much better than  $\tilde{M}_{BPS^*,l}$ ; around 50% less iterations on average for 256 subdomains. It should be mentioned that both methods have the same computational complexity per iteration and that the extra set-up time to build the operator-dependent interpolation (tridiagonal solutions) is negligible.

**4. Concluding remarks.** We proposed a new interpolation operator to define a closely related variant of the BPS\* preconditioner. This operator-dependent interpolation is designed to tackle problems where the PDEs coefficients are either discontinuous or have large variation along the interfaces between the subdomains. The definition of this interpolation is natural on structured meshes with uniform rectangular subdomains and we proposed a generalization to unstructured meshes. This generalization preserves the constant function while taking into account possible discontinuities. The unstructured grid interpolation was inspired by AMGe and also by the fact that it reduces to the original operator dependent interpolation on uniform meshes.

In this paper, we have considered the two dimensional case in order that it be fully understood first. These techniques could be used to modify BPS-like preconditioners for three dimensional problems. The key issue in three dimensions is that the subdomain interfaces are faces and so the one dimensional interpolation must be generalized to two dimensions. This would correspond to taking the elementary stiffness matrices adjacent to the faces and reducing out noninterface unknowns. The operator dependent interpolation is then determined by solving the resulting two dimensional

D-B problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	27	41	35
$\tilde{M}_{BPS^*,od}$	18	17	16
D-Ra problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	31	38	42
$\tilde{M}_{BPS^*,od}$	20	23	23
D-Ri problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	32	44	33
$\tilde{M}_{BPS^*,od}$	13	15	16

TABLE 3.3

# iterations to solve PDEs with discontinuities on unstructured round square mesh (rsq).

D-B problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	22	30	31
$\tilde{M}_{BPS^*,od}$	18	20	12
D-Ra problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	24	40	38
$\tilde{M}_{BPS^*,od}$	21	19	21
D-Ri problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	21	27	32
$\tilde{M}_{BPS^*,od}$	15	17	15

TABLE 3.4

# iterations to solve PDEs with discontinuities on unstructured parallelogram mesh (par).

D-B problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	30	34	37
$\tilde{M}_{BPS^*,od}$	22	21	20
D-Ra problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	31	50	52
$\tilde{M}_{BPS^*,od}$	25	25	35
D-Ri problem			
# subdomains	$4 \times 4$	$8 \times 8$	$16 \times 16$
$\bar{M}_{BPS^*,l}$	28	38	45
$\tilde{M}_{BPS^*,od}$	22	26	22

TABLE 3.5

# iterations to solve PDEs with discontinuities on unstructured round parallelogram mesh (rpar).

PDE discretizations on each face. Given the connections to AMGe which has been developed for three dimensional problems, we anticipate that this would yield an effective method. Unlike AMGe, however, complications associated with applying the algorithm on coarser grids recursively do not arise due to the two-level nature of these domain decomposition preconditioners.

Extensive experiments illustrate the numerical scalability of the BPS\* method on discontinuous coefficient problems even though assumptions used to develop the theory are violated. In practice both linear interpolation as well as operator dependent interpolation yield scalable methods. While both are scalable, however, the operator dependent scheme significantly outperforms the linear interpolation method with similar computational complexity.

#### REFERENCES

- [1] J. BRAMBLE, J. PASCIAK, AND A. SCHATZ, *The construction of preconditioners for elliptic problems by substructuring I.*, Math. Comp., 47 (1986), pp. 103–134.
- [2] M. BREZINA, A. CLEARY, R. FALGOUT, V. HENSON, J. JONES, T. MANTEUFFEL, S. MCCORMICK, AND J. RUGE, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Sci. Comput., 22 (2000), pp. 1570–1592.
- [3] L. CARVALHO, L. GIRAUD, AND P. L. TALLEC, *Algebraic two-level preconditioners for the Schur complement method*, SIAM J. Sci. Comput., 22 (2001).
- [4] T. CHAN AND T. MATHEW, *Domain Decomposition Algorithms*, vol. 3 of Acta Numerica, Cambridge University Press, Cambridge, 1994, pp. 61–143.
- [5] M. DRYJA, B. SMITH, AND O. WIDLUND, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal., 31 (1993), pp. 1662–1694.
- [6] C. FARHAT AND F.-X. ROUX, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Int. J. Numer. Meth. Engng., 32 (1991), pp. 1205–1227.
- [7] L. GIRAUD AND R. S. TUMINARO, *Grid transfer operators for highly variable coefficient problems*, Tech. Rep. TR/PA/93/37, CERFACS, Toulouse, France, 1993. Available on [www.cerfacs.fr/algor/algo\\_reports](http://www.cerfacs.fr/algor/algo_reports).
- [8] A. KLAWONN AND O. B. WIDLUND, *FETI and Neumann-Neumann iterative substructuring methods: connections and new results*, Comm. Pure Appl. Math., LIV (2001), pp. 57–90.
- [9] P. LE TALLEC, *Domain decomposition methods in computational mechanics*, vol. 1 of Computational Mechanics Advances, North-Holland, 1994, pp. 121–220.
- [10] J. MANDEL, *Balancing domain decomposition*, Communications in Numerical Methods in Engineering, 9 (1993), pp. 233–241.
- [11] J. MANDEL AND M. BREZINA, *Balancing domain decomposition for problems with large jumps in coefficients*, Mathematics of Computation, 65 (1996), pp. 1387–1401.
- [12] J. MANDEL AND R. TEZAUER, *Convergence of substructuring method with Lagrange multipliers*, Numer. Math., 73 (1996), pp. 473–487.
- [13] J. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, S. McCormick, ed., Frontiers in Applied Mathematics, Philadelphia, 1987, SIAM, pp. 73–130.
- [14] B. SMITH, *Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity*, PhD thesis, Courant Institute of Mathematical Sciences, Sept. 1990. Tech. Rep. 517, Department of Computer Science, Courant Institute.
- [15] ———, *A domain decomposition algorithm for elliptic problems in three dimensions*, Numer. Math., 60 (1991), pp. 219–234.
- [16] B. SMITH, P. BJØRSTAD, AND W. GROPP, *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, New York, 1st ed., 1996.
- [17] F. G. VASQUEZ, *Internship report on domain decomposition methods for the solution of partial differential equations*, Technical Report TR/PA/00/98, CERFACS, Toulouse, France, 2000. Available on [www.cerfacs.fr/algor/algo\\_reports](http://www.cerfacs.fr/algor/algo_reports).
- [18] W. L. WAN, T. F. CHAN, AND B. SMITH, *An energy-minimizing interpolation for robust multigrid*, SIAM J. Sci. Comput., 21 (2000), pp. 1632 – 1649.
- [19] O. WIDLUND, *Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. Golub, G. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM.