NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2003; **10**:467–484 (DOI: 10.1002/nla.324)

Grid transfer operators for highly variable coefficient problems in two-level non-overlapping domain decomposition methods

L. Giraud[1,*,†], F. Guevara Vasquez[2,‡] and R. S. Tuminaro[3,§]

[1] *CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse Cedex, France*
[2] *CAAM Department, Rice University, 6100 Main St.-MS 134, Houston, TX 77005-1892, U.S.A.*
[3] *Sandia National Laboratories, PO Box 969, Livermore, CA, U.S.A.*

SUMMARY

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 [Bramble *et al.* (Math. Comput. 1986; **47**(175):103)]. 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 behaviour even when the jumps in the coefficients are not aligned with subdomain interfaces. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: 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 *et al.* [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

[*] Correspondence to: L. Giraud, CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse Cedex, France.
[†] E-mail: giraud@cerfacs.fr
[‡] E-mail: fguevara@caam.rice.edu
[§] E-mail: tuminaro@ca.sandia.gov

Contract/grant sponsor: Sandia Corporation, US Department of Energy;
Contract/grant number: DE-AC04-94AL85000.

Published online 7 May 2003
Received 4 July 2001
Revised 18 April 2002

number of mesh points within subdomains. Other non-overlapping domain decomposition preconditioners that possess similar optimality properties include the vertex space [2, 3], the balancing Neumann–Neumann [4–6] and the FETI [7, 8] 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 [9, 4, 10, 11]. 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 [12] and some preliminary results on structured meshes were presented in [13]. 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 [14] 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 robustness 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 behaviour and the robustness of the new preconditioner compared with the BPS* method.

2. STRUCTURED MESHES

We consider the following second order self-adjoint elliptic problem on an open polygonal domain Ω included in $\mathbb{R}^2$:

$$\begin{cases} -\dfrac{\partial}{\partial x}\left(a(x,y)\dfrac{\partial v}{\partial x}\right) - \dfrac{\partial}{\partial y}\left(b(x,y)\dfrac{\partial v}{\partial y}\right) = F(x,y) \quad \text{in } \Omega \\ \hfill v = 0 \quad \text{on } \partial\Omega \end{cases} \tag{1}$$

where $a(x,y), b(x,y) \in \mathbb{R}^2$ are bounded positive functions on Ω. We discretize (1) via finite elements resulting in a sparse symmetric and positive definite (possibly unstructured) matrix equation

$$Au = f$$

We assume that the domain Ω is partitioned into N non-overlapping subdomains $\Omega_1, \ldots, \Omega_N$ with boundaries $\partial\Omega_1, \ldots, \partial\Omega_N$. 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:

$$\begin{pmatrix} A_{II} & A_{IB} \\ A_{IB}^{\mathrm{T}} & A_{BB} \end{pmatrix} \begin{pmatrix} u_I \\ u_B \end{pmatrix} = \begin{pmatrix} f_I \\ f_B \end{pmatrix} \tag{2}$$

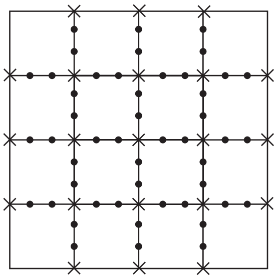

Figure 1. A 4×4 box-decomposition with edge ($\bullet$) and vertex ($\times$) points.

Eliminating u_I from the second block row of (2) leads to the following reduced equation for u_B:

$$Su_B = f_B - A_{IB}^{\mathrm{T}} A_{II}^{-1} f_I, \quad \text{where } S = A_{BB} - A_{IB}^{\mathrm{T}} A_{II}^{-1} A_{IB} \tag{3}$$

is the Schur complement of the matrix A_{II} in A, and is usually referred to as the Schur complement matrix. The reduced system (3) is usually solved via a preconditioned conjugate gradient (PCG) method as the Schur complement matrix inherits the symmetric positive definitiveness 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 $V = \bigcup_i V_i$ be the set with all those indices; each cross point is represented by an $\times$ in Figure 1. We define each edge by considering neighbouring subdomains Ω_j and Ω_l for $j \neq l$, $(j,l) \in \{1,2,\ldots,N\}^2$ where subdomain j and l are neighbouring if $\partial\Omega_j \cap \partial\Omega_l$ is non-empty and contains at least one point not in V. We define each edge E_i by

$$E_i = (\partial\Omega_j \cap \partial\Omega_l) \backslash V$$

In Figure 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

$$B = \left(\bigcup_{i=1}^{m} E_i \right) \cup V \tag{4}$$

consisting of m edges and the cross points.

We now consider BPS-like preconditioners [1]. In this paper, we choose the BPS* method [14]. 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,¶ $R_i = R_{E_i}$, 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

¶ $|.|$ denotes cardinality.

to V. Thus, we set $S_{ij} \equiv R_i S R_j^{\mathrm{T}}$ and $S_V \equiv R_V S R_V^{\mathrm{T}}$. To complete the two-level preconditioner, a coarse grid operator must be defined. Assume that $\Omega_1, \ldots, \Omega_N$ form the elements of a coarse grid mesh, τ^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

$$A_{H,l} = R_l A R_l^{\mathrm{T}} \tag{5}$$

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

$$M_{\mathrm{BPS}*} = \sum_{E_i} R_i^{\mathrm{T}} S_{ii}^{-1} R_i + R_l^{\mathrm{T}} A_{H,l}^{-1} R_l \tag{6}$$

as described in [14]. 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. [15]). This preconditioner (6) can be interpreted as a generalized block Jacobi scheme for the Schur complement system (3) where the block diagonal preconditioning for S_{VV} is omitted and a coarse grid correction is added. The coarse grid term $R_l^{\mathrm{T}} A_{H,l}^{-1} R_l$ incorporates global coupling between distant interfaces. 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

$$\kappa(M_{\mathrm{BPS}} S) = \mathcal{O}(1 + \log^2(H/h)) \tag{7}$$

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

$$\tilde{A}_{H,l} = R_l S R_l^{\mathrm{T}} \tag{8}$$

It has been shown [1, 15] 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 (5) and (8) is ensured. We generally find $\tilde{A}_{H,l}$ more robust when highly discontinuous coefficients are present and so we use (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

$$\tilde{M}_{\mathrm{BPS}*,l} = \sum_{E_i} R_i^{\mathrm{T}} S_{ii}^{-1} R_i + R_l^{\mathrm{T}} \tilde{A}_{H,l}^{-1} R_l \tag{9}$$

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

$$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}} = \sum_{E_i} R_i^{\mathrm{T}} S_{ii}^{-1} R_i + R_{\mathrm{od}}^{\mathrm{T}} \tilde{A}_{H,\mathrm{od}}^{-1} R_{\mathrm{od}} \tag{10}$$

where

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

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, $R_{\mathrm{od}}^{\mathrm{T}}$, 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

$$\begin{cases} -\dfrac{\mathrm{d}}{\mathrm{d}x}(a(x)\dfrac{\mathrm{d}}{\mathrm{d}x}u(x)) = f & \text{in } (0,1) \\ u(x) = 0 & \text{at } x=0 \text{ and } 1 \end{cases} \tag{11}$$

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 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

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

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 ϕ_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 $V^H = \mathrm{span}\{\phi_i^H \in V^h\colon\ i=1,\dots,m\}$ 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 ϕ_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

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

Thus, defining the ϕ_i^H's is equivalent to building a restriction operator, R, and a corresponding interpolation operator, R^{T}.

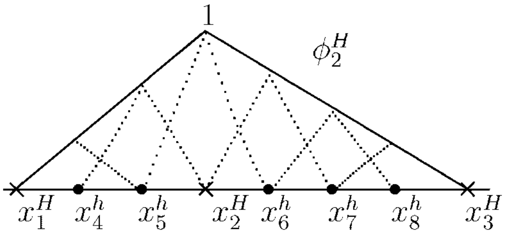

Figure 2. Coarse grid basis function ϕ_2^H associated with the linear interpolation.

We depict in Figure 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 $u \in V_H$, the product $a\dfrac{\partial \tilde{u}}{\partial x}$ is discontinuous at points where $a()$ is discontinuous. Intuitively, the discontinuity of $a\dfrac{\partial \tilde{u}}{\partial x}$ is undesirable as this term is further differentiated in (11). The undesirability of a discontinuous $a\dfrac{\partial \tilde{u}}{\partial x}$ can also be motivated using energy minimization arguments (normally used in algebraic multigrid discussions [16]). 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 $\langle .,.\rangle$ defines the usual vector inner product. However, the discontinuity of $a\dfrac{\partial \tilde{u}}{\partial x}$ 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 to yield a low energy correction. To build an interpolation operator that ensures the continuity of $a\dfrac{\partial \tilde{u}}{\partial x}$ even if $a()$ is discontinuous, we define a new set of coarse grid basis functions (and thus a new grid transfer operator). These are defined by solving the local problems in $[x_{i-1}^H, x_i^H]$:

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(a(x)\,\frac{\mathrm{d}}{\mathrm{d}x}\,\phi_i^H\right) = 0 \quad \text{in } [x_{i-1}^H, x_i^H] \tag{12}$$

with

$$\phi_i^H(x_{i-1}^H) = 0 \quad \text{and} \quad \phi_i^H(x_i^H) = 1$$

In Figure 3, we depict the basis function ϕ_2^H when the function $a()$ is piecewise constant with discontinuities at x_5^h and x_7^h.

The operator-dependent interpolation is numerically constructed by solving the linear system arising from the discretization of (12) on each E_i. That is, interpolation for the two-dimensional problem is built by solving a series of one dimensional problems (one for each subdomain interface). Each one-dimensional horizontal edge problem is constructed by taking $a(x, y)$ in (1) and restricting it (injection) to the proper edge thereby defining $a(x)$ in (12). A similar procedure is performed on vertical edges using $b(x, y)$. While we use injection, $a(x)$ could also be defined by averaging $a(x, y)$ within a neighbourhood of the interface. This averaging would attempt to more accurately capture composite behaviour. We have explored several different averaging procedures and have not found these worthwhile. Averaging usually does

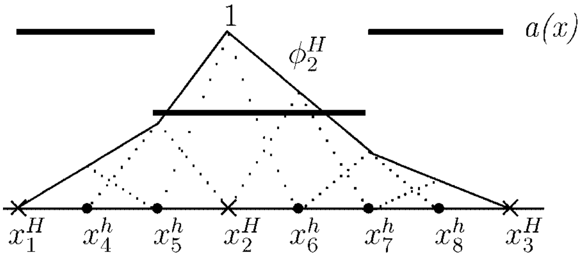

Figure 3. Coarse grid basis function ϕ_2^H associated with the operator-dependent interpolation.

not significantly improve convergence (except on carefully contrived examples) and costs more to compute. For these reasons we do not explore averaging further in this paper.

The discretization of each one-dimensional PDE (12) yields a tridiagonal linear system with the following structure when the Dirichlet boundary conditions are included among the unknowns:

$$A_i = \begin{pmatrix} 1 & 0 & & & \\ \ddots & \ddots & \ddots & & \\ & -a_{\ell-\frac{1}{2}} & (a_{\ell-\frac{1}{2}} + a_{\ell+\frac{1}{2}}) & -a_{\ell+\frac{1}{2}} & \\ & & \ddots & \ddots & \ddots \\ & & & 0 & 1 \end{pmatrix} \qquad (13)$$

where the $(\ell+1)$th row is associated with the ℓth interior vertex on E_i and

$$a_{\ell-\frac{1}{2}} = \int_{x_{\ell-1}^h}^{x_\ell^h} a(x) \nabla \phi_{\ell-1}^h \nabla \phi_\ell^h$$

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

Proof
When $a() \equiv 1$, the solution to (12) such that $\phi_i^H(x_{i-1}^H) = 0$ and $\phi_i^H(x_{i-1}^H) = 1$ 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. □

Lemma 2
The one dimensional operator-dependent interpolation defined by the solution of (12) reduces to the multigrid energy minimization approach [17] 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,

$$R_{\mathrm{od}}^{\mathrm{T}} \mathbf{1}_{\mathbf{V}} = \mathbf{1}_{\mathbf{B}} \tag{14}$$

where the symbol $\mathbf{1}_{\mathbf{V}}$ denotes the vectors of all 1's on the coarse grid and $\mathbf{1}_{\mathbf{B}}$ 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 (12) with $1 \leqslant i \leqslant m$, it follows that

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

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 [17, 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 [4, 5], etc. Connections between this grid transfer and standard operator dependent multigrid transfers can be found in [12].

2.2. Numerical Experiments

To evaluate the sensitivity of the preconditioners to discontinuity, we consider (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

$$\textit{Problem SD-F1}: \ a() = b() = \begin{cases} 1 & \text{in } \mathscr{R}^3 \\ 10^2 & \text{in } \mathscr{R}^2 \cup \mathscr{R}^4 \\ 10^{-2} & \text{in } \mathscr{R}^1 \cup \mathscr{R}^5 \end{cases}$$

$$\textit{Problem SD-F2}: \ a() = b() = \begin{cases} 1 & \text{in } \mathscr{R}^3 \\ 10^3 & \text{in } \mathscr{R}^2 \cup \mathscr{R}^4 \\ 10^{-3} & \text{in } \mathscr{R}^1 \cup \mathscr{R}^5 \end{cases}$$

where the $\mathscr{R}$'s are given by Figure 4. The third problem uses

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

where the $\mathscr{R}$'s are given by Figure 5. For sake of completeness we also consider the classical Poisson problem defined by $a() = b() = 1$.

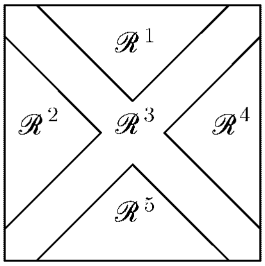

Figure 4. Example 1—Flag.

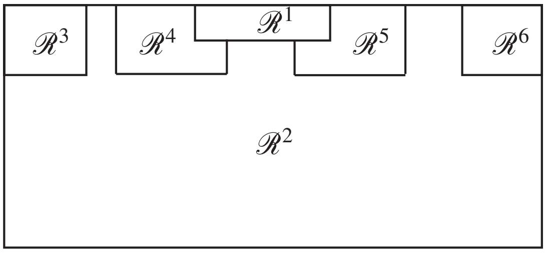

Figure 5. Example 2—Region.

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. The linear systems associated with the coarse space are much smaller than the linear systems associated with the local Dirichlet problems associated with each subdomain. Therefore, we construct the coarse matrix $\tilde{A}_{H,-}$ once, assemble and factor it on all the processors so that we can redundantly perform in parallel its solution via forward/backward substitution at each preconditioned conjugate gradient iteration. One advantage of this redundant calculation is it avoids some communication when solving the coarse problem. More details on the parallel implementation can be found in [18].

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. 256×256 mesh for each subdomain, that is $H/h = 256$). The initial guess x_0 for the conjugate gradient iterations is the null vector and the right hand side is given by a constant vector. In Table I, we give the results for the Poisson problem. As expected, the behaviour of the two variants does not depend on the number of subdomains as predicted by (7) and as indicated by Lemma 1. In Table II 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 $\tilde{M}_{\mathrm{BPS}*,l}$ is less efficient than $\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$. Further,

Table I. Number of iterations to solve the Poisson problem on structured meshes.

No. of subdomains	4 × 4	8 × 8	16 × 16	32 × 32
$\tilde{M}_{\mathrm{BPS}*,l}$	13	17	17	15
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	13	17	17	15

Table II. Number of iterations to solve PDEs with discontinuities on structured meshes.

No. of subdomains	4 × 4	8 × 8	16 × 16	32 × 32
SD-F1 problem				
$\tilde{M}_{\mathrm{BPS}*,l}$	26	33	32	32
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	22	22	27	23
SD-F2 problem				
$\tilde{M}_{\mathrm{BPS}*,l}$	28	36	36	38
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	23	24	31	26
SD-R problem				
$\tilde{M}_{\mathrm{BPS}*,l}$	18	35	37	37
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	17	24	22	19

$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$ is numerically scalable (as is $\tilde{M}_{\mathrm{BPS}*,1}$) and follows a behaviour predicted by (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 (12) cannot be used anymore. However, we can still construct a linear system similar to (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 by reducing out the non-interface unknowns, a tridiagonal matrix is obtained. We depict in Figure 6 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 Neumman 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

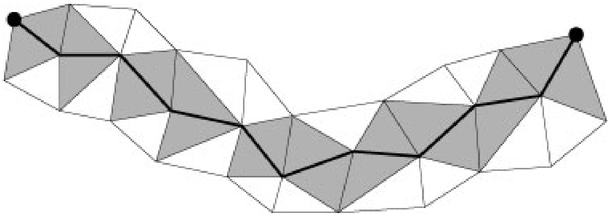

Figure 6. An interface on unstructured meshes.

algorithm [19], the FETI method [7], and the balancing Neumman–Neumman scheme [5]. 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).

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 right-hand 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 non-interface nodes at the elementary level (i.e. before assembling), this elimination could be done after assembling. This would correspond to defining

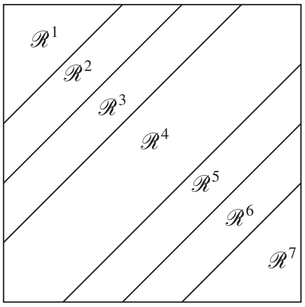

Figure 7. Example 3—Band.

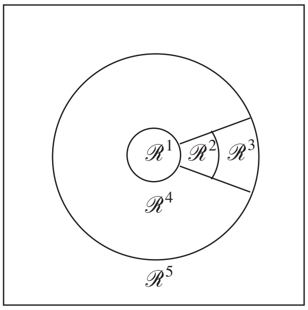

Figure 8. Example 4—Radar.

interpolation as the solution of

$$\begin{cases} -\dfrac{\partial}{\partial x}\left(a(x,y)\,\dfrac{\partial v}{\partial x}\right) - \dfrac{\partial}{\partial y}\left(b(x,y)\,\dfrac{\partial v}{\partial y}\right) = 0 & \text{on } \hat{\Omega} \\ \dfrac{\partial v}{\partial \vec{n}} = 0 & \text{on } \partial\hat{\Omega}\backslash\{V_i, V_j\} \\ v = 1 & \text{at } \{V_i\} \\ v = 0 & \text{at } \{V_j\} \end{cases} \tag{15}$$

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 6). 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 (1) as piecewise constant functions in $(-1,1)^2$ as depicted in Figures 7–9. Using these notations we define a set of

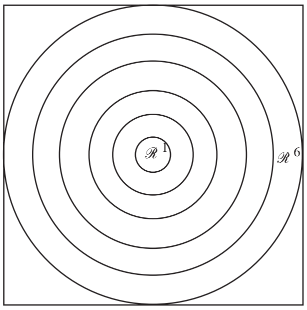

Figure 9. Example 5—Ring.

model problems with discontinuous coefficients as follows:

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

where the $\mathscr{R}^i$ are defined in Figure 7,

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

where the $\mathscr{R}^i$ are defined in Figure 8,

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

where the $\mathscr{R}^i$ are defined in Figure 9. The rings are labelled consecutively starting with the innermost ring labelled as $\mathscr{R}^1$.

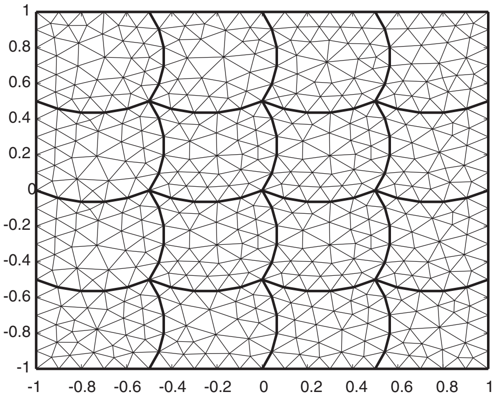

Figure 10. **rsq** unstructured mesh: a 4 × 4 decomposition.

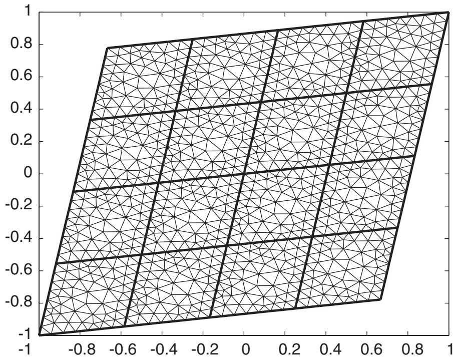

Figure 11. **par** unstructured mesh: a 4 × 4 decomposition.

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 10 (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 11).

rpar Each subdomain is a 'rounded parallelogram' as depicted in Figure 12 (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 composite domain Ω remains the same when the number of subdomains changes.

The mesh generation and the finite element discretization of (1) are done using MATLAB's PDE toolbox. In our implementation, building $\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$ costs a little bit more than constructing $\tilde{M}_{\mathrm{BPS}*,l}$ due to a couple of additional nearest neighbour 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

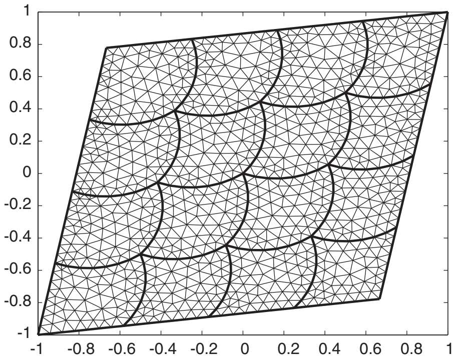

Figure 12. **rpar** unstructured mesh: a 4 × 4 decomposition.

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

Domain decomposition	Approx. range of pts/sd
Parallelograms (**par**)	890–900
Round Squares (**rsq**)	710–860
Round Parallelograms (**rpar**)	780–1100

Table IV. Number of iterations to solve the Poisson problem on unstructured meshes.

No. of subdomains	4 × 4	8 × 8	16 × 16
	Round square mesh (rsq)		
$\tilde{M}_{\mathrm{BPS}*,l}$	16	17	16
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	18	18	17
	Parallelogram mesh (par)		
$\tilde{M}_{\mathrm{BPS}*,l}$	14	14	13
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	14	14	13
	Round parallelogram mesh (rpar)		
$\tilde{M}_{\mathrm{BPS}*,l}$	20	19	18
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	21	21	20

the ratio 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 H/h remains the same. Table III gives the range of number of points per subdomain for each mesh.

We first report in Table IV the numerical behaviour of the two preconditioners for the Poisson problem. It can be observed that for that problem, the two preconditioners have similar convergence behaviour. In addition the convergence is independent of the number of subdomains as predicted by (7). In Tables V–VII we report the number of PCG iterations for $\tilde{M}_{\mathrm{BPS}*,l}$ and $\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$ on discontinuous problems discretized on unstructured meshes. For each

Table V. Number of iterations to solve PDEs with discontinuities on unstructured round square mesh (rsq).

No. of subdomains	4×4	8×8	16×16
	UD-B problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	27	41	35
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	18	17	16
	UD-Ra problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	31	38	42
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	20	23	23
	UD-Ri problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	32	44	33
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	13	15	16

Table VI. Number of iterations to solve PDEs with discontinuities on unstructured parallelogram mesh (par).

No. of subdomains	4×4	8×8	16×16
	UD-B problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	22	30	31
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	18	20	12
	UD-Ra problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	24	40	38
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	21	19	21
	UD-Ri problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	21	27	32
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	15	17	15

Table VII. Number of iterations to solve PDEs with discontinuities on unstructured round parallelogram mesh (rpar).

No. of subdomains	4×4	8×8	16×16
	UD-B problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	30	34	37
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	22	21	20
	UD-Ra problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	31	50	52
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	25	25	35
	UD-Ri problem		
$\tilde{M}_{\mathrm{BPS}*,l}$	28	38	45
$\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$	22	26	22

mesh, we vary the number of subdomains from 16 to 256. As it can be observed, $\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$ performs much better than $\tilde{M}_{\mathrm{BPS}*,l}$; 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.

To simplify the exposition, all of our experiments use the exact S_{ii}^{-1} to define the local components of the preconditioners. For practical reasons, it might be preferable to consider less expensive alternatives. We have performed experiments with an earlier version of this work [12] on structured meshes using an efficient local component [20] computed via a variant of the probing technique [20]. These experiments show that $\tilde{M}_{\mathrm{BPS}*,\mathrm{od}}$ still outperforms $\tilde{M}_{\mathrm{BPS}*,l}$ when an efficient local preconditioner replaces S_{ii}^{-1} [13].

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 non-interface unknowns. The operator dependent interpolation is then determined by solving the resulting two dimensional 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.

ACKNOWLEDGEMENTS

Sandia is a Multiprogram Laboratory Operated by Sandia Corporation, a Lockheed Martin Company, for the US Department of Energy under Contract DE-ACO4-94AL85000.

REFERENCES

1. Bramble JH, Pasciak JE, Schatz AH. The construction of preconditioners for elliptic problems by substructuring I. *Mathematics of Computation* 1986; **47**(175):103–134.
2. Dryja M, Smith BF, Widlund OB. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM Journal on Numerical Analysis* 1993; **31**(6):1662–1694.

3. Smith BF. *Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity*. *PhD thesis*, Courant Institute of Mathematical Sciences, September 1990. Tech. Rep. 517, Department of Computer Science, Courant Institute.
4. Le Tallec P. Domain decomposition methods in computational mechanics, In *Computational Mechanics Advances*, Tinsley Oden J (ed.) vol. 1. North-Holland: Amsterdam, 1994; 121–220.
5. Mandel J. Balancing domain decomposition. *Communications in Numerical Methods in Engineering* 1993; **9**:233–241.
6. Mandel J, Brezina M. Balancing domain decomposition for problems with large jumps in coefficients. *Mathematics of Computation* 1996; **65**:1387–1401.
7. Farhat C, Roux F-X. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering* 1991; **32**:1205–1227.
8. Mandel J, Tezaur R. Convergence of substructuring method with Lagrange multipliers. *Numerische Mathematik* 1996; **73**:473–487.
9. Klawonn A, Widlund OB. FETI and Neumann–Neumann iterative substructuring methods: connections and new results. *Communications in Pure and Applied Mathematics* 2001; **LIV**:57–90.
10. Smith BF. A domain decomposition algorithm for elliptic problems in three dimensions. *Numerische Mathematik* 1991; **60**:219–234.
11. Widlund OB. Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane. In Glowinski R, Golub GH, Meurant GA, Périaux J (eds). *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM: Philadelphia, PA, 1988.
12. Giraud L, Tuminaro RS. Grid transfer operators for highly variable coefficient problems. *Technical Report TR/PA/93/37*, CERFACS, Toulouse, France, 1993. Available on `www.cerfacs.fr/algor/algo_reports`.
13. Carvalho LM, Giraud L, Le Tallec P. Algebraic two-level preconditioners for the Schur complement method. *SIAM Journal on Scientific Computing* 2001; **22**(6):1987–2005.
14. Chan TF, Mathew TP. Domain Decomposition Algorithms. In *Acta Numerica*, Iserles A (ed.) vol. 3. Cambridge University Press: Cambridge, 1994; 61–143.
15. Smith BF, Bjørstad P, Gropp W. *Domain Decomposition*, *Parallel Multilevel Methods for Elliptic Partial Differential Equations*. 1st edn. Cambridge University Press: New York, 1996.
16. Wan WL, Chan TF, Smith B. An energy-minimizing interpolation for robust multigrid. *SIAM Journal on Scientific Computing* 2000; **21**(4):1632–1649.
17. Guevara Vasquez F. 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`.
18. Brezina M, Cleary A, Falgout R, Henson V, Jones J, Manteuffel T, McCormick S, Ruge J. Algebraic multigrid based on element interpolation (AMGe). *SIAM Journal on Scientific Computing* 2000; **22**(5):1570–1592.
19. Giraud L, Tuminaro RS. Schur complement preconditioners for anisotropic problems. *IMA Journal of Numerical Analysis* 1999; **19**(1):1–17.
20. Chan TF, Mathew TP. The interface probing technique in domain decomposition. *SIAM Journal on Matrix Analysis and Applications* 1992; **13**:212–238.