UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Fourier and Probe Variants of the Vertex Space Domain Decomposition Algorithm

Tony F. Chan
Tarek P. Mathew
Jian-Ping Shao

January 1992

CAM Report 92-06

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

FOURIER AND PROBE VARIANTS OF THE VERTEX SPACE DOMAIN DECOMPOSITION ALGORITHM

January 28, 1992

TONY F. CHAN , TAREK P. MATHEW AND JIAN-PING SHAO *

Abstract. The vertex space algorithm of Smith [18] is a domain decomposition method for two dimensional elliptic problems based on non-overlapping subregions, in which the reduced Schur complement system on the interface is solved using a generalized block Jacobi type preconditioner, with the blocks corresponding to the vertex space, edges and a coarse grid. In this paper, we describe several variants of this algorithm derived from using two kinds of approximations for the edge and vertex space sub-blocks, one based on Fourier approximation, and another based on an algebraic *probing* technique in which sparse approximations to these sub-blocks are computed. Our motivation is to improve efficiency of the algorithm without sacrificing the optimal convergence rate. Numerical and theoretical results on the performance of these algorithms are presented.

Key Words. Domain decomposition, schur complement, interface probe, block Jacobi preconditioner, elliptic equations, preconditioners, vertex spaces.

AMS subject classifications: 65N20, 65F10.

1. Introduction. In this paper, we describe efficient versions of two domain decomposition algorithms based on non-overlapping subregions for solving self adjoint elliptic problems in two dimensions. This paper is a shortened version of [8], where additional numerical results and proofs of some of the results stated here are presented. The algorithms we describe are variants of the vertex space algorithm (VS) proposed by Smith [18] and Nepomnyaschikh [17], and an algorithm of Bramble, Pasciak and Schatz (BPS) [2], both of which can be viewed as block Jacobi type preconditioners for solving the reduced Schur complement system on the interface.

In the original version of the VS preconditioner [18], the sub-blocks of the Schur complement, which are dense matrices, are computed and inverted using direct methods. In order to reduce this overhead cost, we consider using approximations which are inexpensive to construct, and inexpensive to invert. Two kinds of approximations are considered, one based on Fourier approximations of the interface operators, and another based on sparse algebraic approximation of the interface operators by a *probing* technique. The Fourier based approximations yield spectrally equivalent preconditioners with respect to mesh size variations. However, their performance can be sensitive to the coefficients. On the other hand, the probing based algorithms adapt well to the coefficients, but can be sensitive to mesh size variations.

In § 2, we describe the elliptic problem and the Schur complement system on the interface. In § 3, we describe the original versions of the BPS and VS preconditioners for the Schur complement on the interface. In § 4, we describe the Fourier and *probing* variants of the BPS and VS preconditioners. Finally, in § 5, we present numerical results comparing the rates of convergence of the various preconditioners.

* Department of Mathematics, University of California at Los Angeles, Los Angeles, CA. 90024.
This work was supported in part by the Department of Energy under contract DE-FG03-87ER25037, by the Army Research Office under contract ARO DAAL03-91-G-150, by the National Science Foundation under grant FDP NSF ASC 9003002, and by the Office for Naval Research under contract ONR N00014-90-J-1695. *This paper is to appear in the Proceedings of the 5th Domain Decomposition Conference held at Norfolk, Virginia, in May 1991, SIAM, Philadelphia.*

2. Non-overlapping domain decomposition approach. We consider the following 2nd order self adjoint elliptic problem on a polygonal domain $\Omega \in R^2$:

$$(1) \qquad \begin{cases} -\nabla \cdot (a(x,y)\nabla u) &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega, \end{cases}$$

where $a(x,y) \in R^{2\times 2}$ is a symmetric, uniformly positive definite matrix function having $L^\infty(\Omega)$ entries, and $f \in L^2(\Omega)$.

We assume that the domain Ω is partitioned into N non-overlapping subdomains $\Omega, \cdots, \Omega_N$ of diameter H, which form the elements of a *quasi-uniform* coarse grid triangulation τ^H, see Fig. 1. We also assume that the subdomains Ω_i are refined to produce a fine grid *quasi-uniform* triangulation τ^h having elements of diameter h. Corresponding to the coarse grid and fine grid triangulations, we discretize (1) either by using finite elements, see [10], or by using finite difference methods, see [19], resulting in a symmetric positive definite linear system $A_h u_h = f_h$, on the fine grid and $A_H u_H = f_H$, on the coarse grid.

Let I denote the union of the interiors of the subdomains, and let B denote the interface separating the subdomains, i.e. $I = \cup_i \Omega_i, \quad B \equiv (\cup_i \partial\Omega_i) - \partial\Omega$. Then, grouping the unknowns in I in the vector u_I and the unknowns on B in the vector u_B, we obtain a reordering of the fine grid problem:

$$(2) \qquad \begin{bmatrix} A_{II} & A_{IB} \\ A_{IB}^T & A_{BB} \end{bmatrix} \begin{bmatrix} u_I \\ u_B \end{bmatrix} = \begin{bmatrix} f_I \\ f_B \end{bmatrix}.$$

Here A_{II} corresponds to the coupling between nodes in the interior of the subdomains. For standard discretizations $A_{II} \equiv blockdiag(A_{11}, \cdots, A_{NN})$ is a block diagonal matrix.

Eliminating the interior unknowns u_I, we obtain $u_I = A_{II}^{-1}(f_I - A_{IB}u_B)$, and substituting this in the 2nd block row of (2) yields an equation for u_B:

$$(3) \qquad S u_B = f_B - A_{IB}^T A_{II}^{-1} f_I,$$

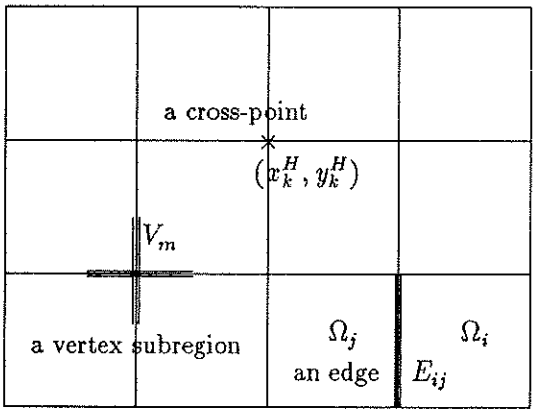
where $S = A_{BB} - A_{IB}^T A_{II}^{-1} A_{IB}$ is referred to as the Schur complement or interface matrix.

We consider solving (3) by a preconditioned iterative method such as the conjugate gradient method, see [14], without the explicit construction of S. In this case only matrix vector products with S are required, and each such matrix vector product requires the solution of one problem on each subdomain Ω_i. The Schur complement, however, is ill-conditioned with $\kappa(S) \approx O(h^{-1})$, see [1, 2], and therefore requires a preconditioner M; the construction of efficient preconditioners M for S will be the main focus of this paper.

3. The BPS and VS preconditioners for S. The Bramble, Pasciak and Schatz preconditioner (BPS) [2], and the vertex space preconditioner (VS) of Smith [18] and Nepomnyaschikh [17] can both be interpreted as generalized block Jacobi type preconditioners for (3) with overlapping blocks and involving residual correction on a coarse grid. Variants of these preconditioners will be discussed in § 4.

3.1. Notations for a partition of the interface B. In the case of many subdomains, the interface B can be partitioned as a union of edges E_{ij} and cross-points V, see Fig. 1: $B = \cup_{ij} E_{ij} \quad \cup V$, where E_{ij} denotes the edge separating subdomains Ω_i and

FIG. 1. *The vertex space partitioning of the interface.*



Ω_j, and V denotes the collection of cross-points (vertices (x_k^H, y_k^H) of the subdomains). Note that the edges E_{ij} are assumed not to include its endpoints.

For each edge E_{ij} we define $R_{E_{ij}}$ as the standard pointwise restriction of nodal values to E_{ij}, see [8]. Its transpose $R_{E_{ij}}^T$ extends grid functions in E_{ij} by zero to the rest of B. Similarly, we define R_V as the standard pointwise restriction map onto the cross-points, and its transpose R_V^T is thus extension by zero to B of nodal values in V.

3.2. The BPS preconditioner. The BPS preconditioner can be viewed as a modification of the standard block Jacobi preconditioner M_J, consisting of diagonal blocks of the Schur complement S in the block partitioning of the interface B into the cross-points V and n edges E_{ij}, in some ordering $E_1, \cdots, E_n$ of the edges:

$$S = \begin{bmatrix} S_{E_1} & \cdots & S_{E_1E_n} & S_{E_1V} \\ \vdots & \cdots & \vdots & \vdots \\ S_{E_1E_n}^T & \cdots & S_{E_n} & S_{E_nV} \\ S_{E_1V}^T & \cdots & S_{E_nV}^T & S_V \end{bmatrix}.$$

Here, $S_{E_iE_j} \equiv R_{E_i} S R_{E_i}^T$ denotes the coupling in S between nodes on E_i and E_j, and $S_{E_iV} \equiv R_{E_i} S R_V^T$ denotes the coupling in S between nodes on E_i and V. Note that edges E_i and E_j will be coupled in S only if they are part of the boundary of a common subdomain Ω_k. S is thus a block sparse matrix, and corresponding to each edge E_{ij}, the submatrix $S_{E_{ij}}$ is identical to the two subdomain Schur complement on interface E_{ij} separating Ω_i and Ω_j. The submatrix S_V, which corresponds to coupling in S between cross-points, is almost a diagonal matrix since the cross points are weakly coupled in S. In the case of five point discretizations on rectangular subdomains, S_V is diagonal.

Corresponding to this block partition of S, the block Jacobi preconditioner M_J is defined by:

$$M_J^{-1} g_B = \sum_{\text{edges } ij} R_{E_{ij}}^T S_{E_{ij}}^{-1} R_{E_{ij}} f_B + R_V^T S_V^{-1} R_V f_B. \tag{4}$$

This block Jacobi preconditioned system can be shown to have an $O(H^{-2})$ condition

number, see [2, 20]. This can be attributed to the absence of global communication of information amongst all the edges in the preconditioning step.

The original version of the BPS algorithm [2] involves two changes to this block Jacobi preconditioner. One is that the submatrices $S_{E_{ij}}$ are replaced by Fourier based approximations $\tilde{S}_{E_{ij}}$ which will be described in § 4. The second change is to incorporate global coupling by replacing the cross-points correction term $R_V^T S_V^{-1} R_V$ by the standard coarse grid correction term $R_H^T A_H^{-1} R_H$ as in two level multigrid methods, (involving weighted restriction and interpolation maps R_H and R_H^T respectively,) see [8]. The BPS preconditioner is defined by:

$$M_{BPS}^{-1} f_B = \sum_{\text{edges } i,j} R_{E_{ij}}^T \tilde{S}_{E_{ij}}^{-1} R_{E_{ij}} f_B + R_H^T A_H^{-1} R_H f_B,$$

and the resulting condition number is improved over that of the block Jacobi version.

THEOREM 3.1. *The BPS preconditioner satisfies*

$$\frac{\lambda_{max}(M_{BPS}^{-1} S)}{\lambda_{min}(M_{BPS}^{-1} S)} \leq c_2(1 + \log^2(H/h)),$$

where c_2 is independent of H and h.

Proof. See [2] and [20]. □

3.3. The vertex space algorithm of Smith and Nepomnyaschikh. The logarithmic growth in the condition number of the BPS preconditioner can be attributed to the neglect of coupling between adjacent edges of B. The VS preconditioner of Smith [18] and Nepomnyaschikh [17] incorporates some coupling between adjacent edges through the use of certain overlapping blocks of S corresponding to nodes on certain *vertex regions* V_k, which will be defined, and it leads to a condition number independent of mesh parameters.

Let V_k denote the portion of B within a distance of βH from (x_k^H, y_k^H) for some positive fraction $0 < \beta < 1$, see Fig. 1. We refer to each V_k as a vertex region or vertex space. Corresponding to each vertex region, we denote the pointwise restriction map onto V_k by R_{V_k}, where, for any grid function g_B on B, we have $R_{V_k} g_B = g_B$ on V_k. Its transpose $R_{V_k}^T$ is thus extension by zero outside V_k. The submatrix S_{V_k} corresponding to V_k is defined by $S_{V_k} = R_{V_k} S R_{V_k}^T$. The action of the inverse of the vertex space preconditioner M_{VS} involves the inversion of these new overlapping blocks in addition to the blocks used in the BPS preconditioner:

$$(5) \quad M_{VS}^{-1} f_B = R_H^T A_H^{-1} R_H f_B + \sum_{E_{ij}} R_{E_{ij}}^T (S_{E_{ij}})^{-1} R_{E_{ij}} f_B + \sum_{V_k} R_{V_k}^T (S_{V_k})^{-1} R_{V_k} f_B.$$

The following result is proved in [18, 17].

THEOREM 3.2. *Suppose the overlap of the vertex regions V_k is βH, then:*

$$\frac{\lambda_{max}(M_{VS}^{-1} S)}{\lambda_{min}(M_{VS}^{-1} S)} \leq C(\beta),$$

where $C(\beta)$ is independent of H and h.

4. Two variants of the vertex space method. The implementation of the VS algorithm requires the computation of the edge and vertex matrices $S_{E_{ij}}$ and S_{V_k}. If there are n_i nodes on each $\partial\Omega_i \cap B$, then computing all the submatrices $S_{E_{ij}}$ and S_{V_k} would require solving n_i problems on each Ω_i. This expense can be significantly reduced if the exact edge and vertex matrices are replaced by approximations which can be computed and inverted at significantly less cost. Thus the modified VS algorithms that we will derive have the form:

$$M_{MVS}^{-1} \equiv R_H^T A_H^{-1} R_H + \sum_{ij} R_{E_{ij}}^T (\tilde{S}_{E_{ij}})^{-1} R_{E_{ij}} + \sum_k R_{V_k}^T (\tilde{S}_{V_k})^{-1} R_{V_k}, \tag{6}$$

where $\tilde{S}_{E_{ij}}$ and $\tilde{S}_{V_k}$ are approximations to $S_{E_{ij}}$ and S_{V_k} respectively, and the corresponding modified BPS algorithm is:

$$M_{MBPS}^{-1} \equiv R_H^T A_H^{-1} R_H + \sum_{ij} R_{E_{ij}}^T (\tilde{S}_{E_{ij}})^{-1} R_{E_{ij}}. \tag{7}$$

4.1. Fourier approximations. Fourier based approximations of the edge and vertex matrices are constructed based on the property that, restricted to simple curves the Schur complement is spectrally equivalent to the square root of the Laplace operator on it, and this has been studied extensively, see [1, 13, 4, 3, 12, 5].

For the edge approximation, we use:

$$\tilde{S}_{E_{ij}}^F \equiv D_{ij}^{1/2} W diag(\mu_k) W D_{ij}^{1/2}, \tag{8}$$

where D_{ij} denotes the diagonal of A_h restricted to E_{ij}, $\mu_k = \sqrt{\lambda_k(1 - \frac{\lambda_k}{6})}$ with $\lambda_k = 4\sin^2(\frac{k\pi h}{2})$ and $W_{ij} = \sqrt{2h}\sin(ij\pi h)$. We note that this is a diagonally scaled form of the approximation first used by [2]. The scaling is added to increase adaptivity to the variation in the coefficient $a(x, y)$ in (1).

Next, we describe approximations of the vertex space matrices S_{V_k} based on Fourier techniques for the model geometry of Fig. 3. Let u_{V_k} be a grid function on B which is zero outside the vertex region V_k, i.e., zero on $B - V_k$. Then, by a property of the Schur complement [8], we obtain that

$$u_{V_k}^T S_{V_k} u_{V_k} = \sum_{i=1}^{4} u_{V_k}^T S^{(i)} u_{V_k}, \tag{9}$$

where $S^{(i)}$ is the component of the Schur complement originating from Ω_i. For $i = 1, 2, 3, 4$, let L_i^k denote the L-shaped segment $V_k \cap \partial\Omega_i$, and further let $R_{L_i^k}$ denote the pointwise restriction onto L_i^k. Then, as in the case for the edges, $(R_{L_i^k} u_B)^T S^{(i)} (R_{L_i^k} u_B)$, is spectrally equivalent to $(R_{L_i^k} u_B)^T M_i^k (R_{L_i^k} u_B)$ where M_i^k is any of the known unscaled Fourier approximations to the square root of the Laplacian on L_i^k (see e.g. [9, 8].) Let D_i^k denote the diagonal of $A^{(i)}$ restricted to L_i^k. We define:

$$\tilde{S}_{V_k}^F \equiv \sum_{i=1}^{4} R_{L_i^k}^T (D_i^k)^{1/2} M_i^k (D_i^k)^{1/2} R_{L_i^k}. \tag{10}$$

Note that each term in the above sum is a small dense matrix which can be computed inexpensively. By construction, the matrices $\tilde{S}_{V_k}^F$ are spectrally equivalent to S_{V_k}, from

which we can easily prove the following result for the combined Fourier preconditioner M_{FVS} using both $\tilde{S}^F_{E_{ij}}$ and $\tilde{S}^F_{V_k}$ [8]:

THEOREM 4.1. *The Fourier preconditioner M_{FVS} satisfies:*

$$c_0 \leq \frac{\lambda_{max}(M^{-1}_{FVS}S)}{\lambda_{min}(M^{-1}_{FVS}S)} \leq c_1,$$

where c_0, c_1 are independent of H, h, but may depend on the overlap ratio β. For most applications we considered, it was sufficient to choose the number of nodes on the vertex regions V_k to be small, say 5 or 9, and so the matrices $\tilde{S}^F_{V_k}$ can be computed and inverted at little expense by direct methods.

4.2. Probe approximations.

4.2.1. Edge probe approximations. In its basic form, the *probing* technique [9, 15, 16, 11] consists of approximating each $S_{E_{ij}}$ by a tridiagonal matrix $\tilde{S}_{E_{ij}}$ which is chosen on the assumption that each node on an edge is strongly coupled in S only to nodes adjacent to it and weakly coupled to the other nodes.

To obtain a tridiagonal approximation $\tilde{S}_{E_{ij}}$ to $S_{E_{ij}}$, we equate the matrix vector products $S_{E_{ij}}p_i$ to $\tilde{S}_{E_{ij}}p_i$ for the following three *probe* vectors p_i:

$$p_1 = [1,0,0,1,0,0,\cdots]^T, \quad p_2 = [0,1,0,0,1,0,\cdots]^T, \quad p_3 = [0,0,1,0,0,1,\cdots]^T,$$

resulting in:

$$\begin{bmatrix} (\tilde{S}_{E_{ij}})_{11} & (\tilde{S}_{E_{ij}})_{12} & 0 \\ (\tilde{S}_{E_{ij}})_{21} & (\tilde{S}_{E_{ij}})_{22} & (\tilde{S}_{E_{ij}})_{23} \\ (\tilde{S}_{E_{ij}})_{34} & (\tilde{S}_{E_{ij}})_{32} & (\tilde{S}_{E_{ij}})_{33} \\ \vdots & \vdots & \vdots \end{bmatrix} := [S_{E_{ij}}p_1, S_{E_{ij}}p_2, S_{E_{ij}}p_3], \tag{11}$$

from which the non-zero entries of $\tilde{S}_{E_{ij}}$ can be easily read off. In general, $\tilde{S}_{E_{ij}}$ will not preserve the symmetry of $S_{E_{ij}}$, and so we symmetrize it to obtain $\tilde{S}^P_{E_{ij}}$ using the following *minimum-modulus* procedure [7]:

$$(\tilde{S}^P_{E_{ij}})_{ij} \equiv \begin{cases} (\tilde{S}_{E_{ij}})_{ji} & \text{if } |(\tilde{S}_{E_{ij}})_{ji}| \leq |(\tilde{S}_{E_{ij}})_{ij}| \\ (\tilde{S}_{E_{ij}})_{ij} & \text{if } |(\tilde{S}_{E_{ij}})_{ij}| \leq |(\tilde{S}_{E_{ij}})_{ji}|. \end{cases}$$

We will denote the construction of $\tilde{S}^P_{E_{ij}}$ from $S_{E_{ij}}p_1, S_{E_{ij}}p_2, S_{E_{ij}}p_3$ by the notation:

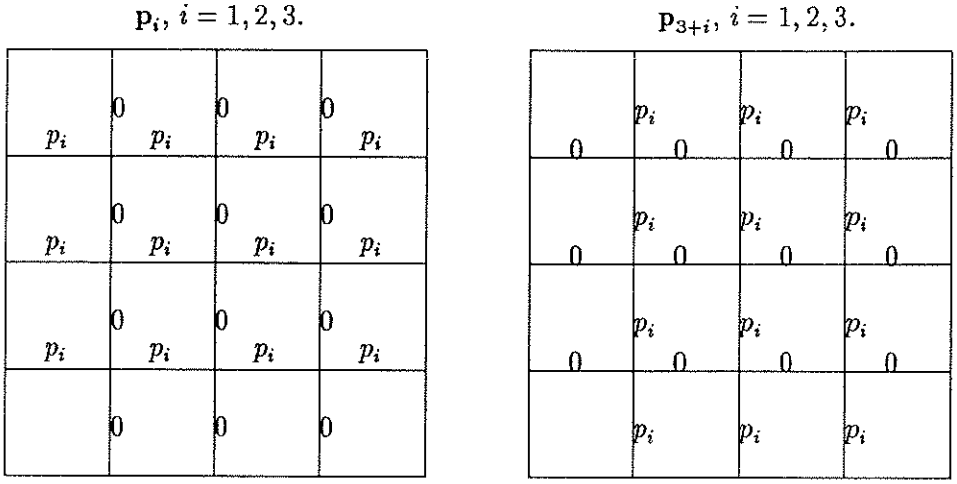
$$\tilde{S}^P_{E_{ij}} = \text{PROBE}(S_{E_{ij}}p_1, S_{E_{ij}}p_2, S_{E_{ij}}p_3). \tag{12}$$

It can be shown [7] that $\tilde{S}^P_{E_{ij}}$ preserves row-wise diagonal dominance of $S^P_{E_{ij}}$.

Computing the three matrix vector products $S_{E_{ij}}p_i$ requires three solves on each subdomain Ω_i and Ω_j. Thus, in order to compute edge approximations $\tilde{S}^P_{E_{ij}}$ on the edges of all the subdomains, twelve solves on each subdomain would be required, since the boundary of rectangular subdomains consists of four edges.

We now describe a procedure for computing all the edge approximations using only six solves on each subdomain, by simultaneously prescribing boundary conditions on other edges, an idea first used in Keyes and Gropp [15, 16]. To minimize the approximation errors arising from the coupling between vertical and horizontal edges,

FIG. 2. *Simultaneous probe vectors*



we will specify probe vectors p_i either on all horizontal edges simultaneously, or on all vertical edges simultaneously. For $i = 1, 2, 3$, see Fig. 2, define:

$$\mathbf{p}_i \equiv \begin{cases} p_i & \text{on all horizontal edges} \\ 0 & \text{on all vertical edges} , \end{cases} \qquad \mathbf{p}_{3+i} \equiv \begin{cases} 0 & \text{on all horizontal edges} \\ p_i & \text{on all vertical edges.} \end{cases}$$

On the horizontal edges, the probe vectors p_i can be ordered from left to right, and on vertical edges from bottom to top. For these six probe vectors, we compute the discrete harmonic extensions $E^h\mathbf{p}_i = (-A_{II}^{-1}A_{IB}\mathbf{p}_i, \mathbf{p}_i)$, and this involves six solves on each subdomain. If E_{ij} is an horizontal edge, we define:

$$\tilde{S}^P_{E_{ij}} = \text{PROBE}(R_{E_{ij}}A_hE^h\mathbf{p}_1, R_{E_{ij}}A_hE^h\mathbf{p}_2, R_{E_{ij}}A_hE^h\mathbf{p}_3).$$

If E_{ij} is a vertical edge, then we define:

$$\tilde{S}^P_{E_{ij}} = \text{PROBE}(R_{E_{ij}}A_hE^h\mathbf{p}_4, R_{E_{ij}}A_hE^h\mathbf{p}_5, R_{E_{ij}}A_hE^h\mathbf{p}_6).$$

THEOREM 4.2. *If the coefficient matrix A_h for the model rectangular geometry of Fig. 1 satisfies the discrete strong maximum principle (as is the case for standard five point discretizations), then the probe approximations $\tilde{S}^P_{E_{ij}}$ obtained above are strictly diagonally dominant.*

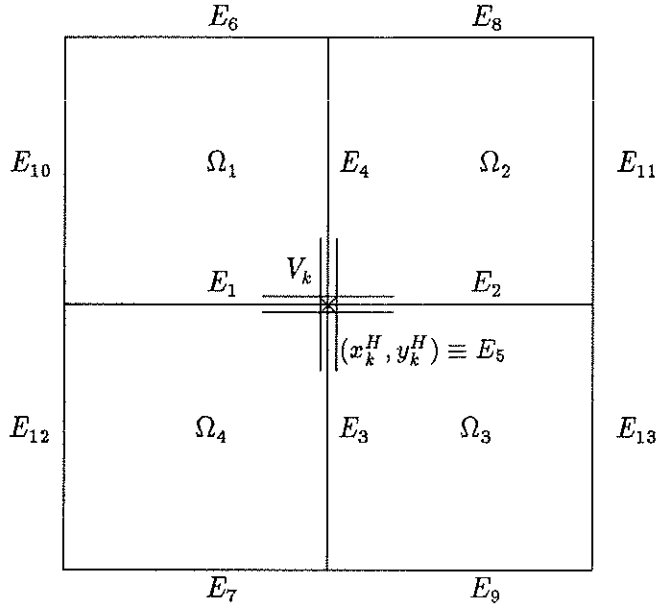
Proof. See [8] for details. □

4.2.2. Probe vertex approximations. Next, we describe how to adapt the probing technique to construct sparse approximations to the vertex submatrices S_{V_k}. We use a slight modification of a technique first described in [6], which allows us to prove that the approximation is non-singular and preserves diagonal dominance.

For simplicity, we will describe this procedure for the vertex region V_k in the center of the subdomains $\Omega_1, \cdots, \Omega_4$ of Fig. 3. We partition V_k into five disjoint regions:

$$V_k = (V_k \cap E_1) \cap (V_k \cap E_2) \cap (V_k \cap E_3) \cap (V_k \cap E_4) \cap (x_k^H, y_k^H), \tag{13}$$

FIG. 3. *Numbering of Edges.*



and we obtain a corresponding 5×5 block partition of the vertex matrix S_{V_k}:

$$S_{V_k} = \begin{bmatrix} S_{11} & 0 & S_{13} & S_{14} & S_{15} \\ 0 & S_{22} & S_{23} & S_{24} & S_{25} \\ S_{13}^T & S_{23}^T & S_{33} & 0 & S_{35} \\ S_{14}^T & S_{24}^T & 0 & S_{44} & S_{45} \\ S_{15}^T & S_{25}^T & S_{35}^T & S_{45}^T & S_{55} \end{bmatrix},$$

where each S_{ij} corresponds to the coupling between nodes in block i and block j. The submatrices S_{12} and S_{34} and their transposes are zero, since there is no coupling in S between nodes in E_1 and E_2, and between nodes in E_3 and E_4. We will construct a vertex matrix approximation $\tilde{S}_{V_k}^P$ having the same block structure as S_{V_k}, with sub-blocks $\tilde{S}_{ij}$ which will be chosen to be sparse.

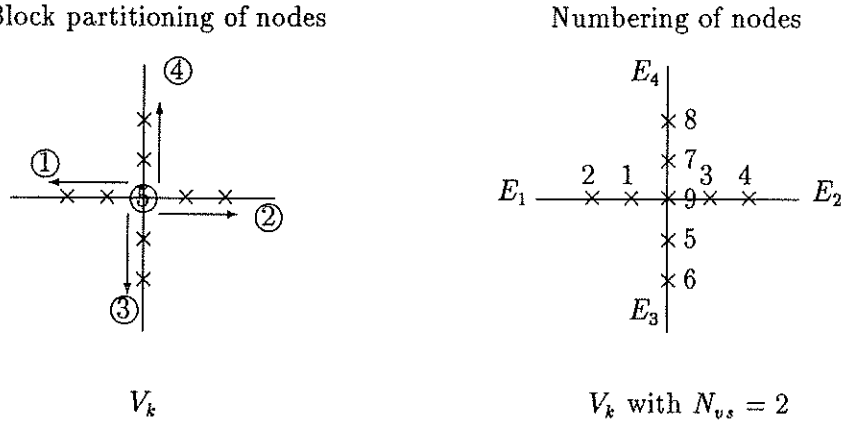
To facilitate description of the sparsity pattern, we will use the following ordering of nodes within V_k; for each of the four edge segments $E_i \cap V_k$, the nodes will be numbered to increase away from the cross-point (x_k^H, y_k^H), which is ordered last. This ordering is shown in Fig. 4 when each segment $E_i \cap V_k$ contain just two nodes.

Our choice of the sparsity pattern for the sub-blocks $\tilde{S}_{ij}$ is based on the assumption that the elements of S_{V_k} decay with increasing distance between nodes.

Definition and computation of the edge blocks $\tilde{S}_{ii}$ for $i = 1, 2, 3, 4$. Within each edge segment $E_i \cap V_k$ we assume the coupling in S_{V_k} is strong only between adjacent nodes. Based on this assumption, S_{ii} will be approximated by tridiagonal matrices $\tilde{S}_{ii}$ which are chosen to be the submatrices of the tridiagonal edge matrices $\tilde{S}_{E_i}^P$ for $i = 1, 2, 3, 4$, which were computed in § 4.2.

Definition and computation of the blocks $\tilde{S}_{i5}$ for $i = 1, \cdots, 5$. We assume the cross-point (x_k^H, y_k^H) is coupled strongly in S_{V_k} only to the nodes adjacent it. Based on this assumption, we choose the vectors $\tilde{S}_{i5}$ to be zero except in the first entry, i.e., $\tilde{S}_{i5} = ((\tilde{S}_{i5})_1, 0, \cdots, 0)^T$, for $i = 1, \cdots, 5$. For five point discretizations on the rectangular geometry of Fig. 1, it can easily be shown that the last row and column of

FIG. 4. *Ordering of unknowns within each vertex subregion V_k*



S_{V_k} is exactly equal to the last row and column of $R_{V_k} A_h R_{V_k}^T$, the matrix A_h restricted to V_k. Therefore, we define

$$\tilde{S}_{i5} \equiv A_{i5} = S_{i5}, \quad i = 1, \cdots, 5, \quad \text{and} \quad \tilde{S}_{5i} \equiv A_{5i} = S_{5i}, \quad i = 1, \cdots, 5.$$

Definition and computation of $\tilde{S}_{ij}$ for $i = 1, 2$ and $j = 3, 4$. We assume the couplings in S_{V_k} between edge segments $E_i \cap V_k$ and $E_j \cap V_k$ is strong only between the nodes which are closest (adjacent) to the cross-point (x_k^H, y_k^H). Based on this assumption, we choose the submatrices $\tilde{S}_{13}$, $\tilde{S}_{14}$, $\tilde{S}_{23}$ and $\tilde{S}_{24}$ and their transposes to have all zero entries except for the $(1,1)$-th entry. So there are only eight non-zero entries to define.

Consider for example the entry $(\tilde{S}_{14})_{11}$, which we would like to be an approximation to $(S_{14})_{11}$, the coupling in S between node $(x_k^H - h, y_k^H)$ and node $(x_k^H, y_k^H + h)$. Note that $(S_{14})_{11} = (S\delta_k)(x_k^H - h, y_k^H)$ (i.e. the component of $S\delta_k$ corresponding to the point $(x_k^H - h, y_k^H)$) where δ_k is the boundary data which is 1 on $(x_k^H, y_k^H + h)$ and zero elsewhere, and therefore computing $(S_{14})_{11}$ requires one subdomain solve. In order to reduce this overhead, we would like to extract an approximation from the subdomain solves we already used for the probe edge approximations. For example, one could define $(\tilde{S}_{14})_{11} = (S\mathbf{p}_4)(x_k^H - h, y_k^H)$. However, it turns out that this definition can lead to a non-diagonally dominant (and possibly singular) $\tilde{S}_{V_k}$. This can be seen by noting that

$$(S\mathbf{p}_4)(x_k^H - h, y_k^H) = (S_{E_1E_4}p_1 + S_{E_1E_{10}}p_1 + S_{E_1E_3}p_1 + S_{E_1E_{12}}p_1)(x_k^H - h, y_k^H).$$

The last two terms on the right corresponds to extra influence from Ω_4 on the coupling between nodes $(x_k^H - h, y_k^H)$ and $(x_k^H, y_k^H + h)$ (which should only involve couplings within Ω_1). These extra couplings could cause loss of diagonal dominance, since, in case the coefficients are large in Ω_4, the last two terms will dominate the sum on the right. In order to eliminate the influence from Ω_4, we now define

$$(\tilde{S}_{14})_{11} = (S_{E_1E_4}p_1 + S_{E_1E_{10}}p_1)(x_k^H - h, y_k^H) \quad \left(\equiv (R_{E_1}A^{(1)}E^h\mathbf{p}_4)_1\right),$$

FIG. 5. *Discontinuos coefficients* $a(x,y)$

$a = 300$	$a = 10^{-4}$	$a = 31400$	$a = 5$
$a = 0.05$	$a = 6$	$a = 0.07$	$a = 2700$
$a = 10^{6}$	$a = 0.1$	$a = 200$	$a = 9$
$a = 1$	$a = 6000$	$a = 4$	$a = 140000$

where we recall that $A^{(1)}$ is the local stiffness matrix on Ω_1. The last equality comes from the definition of the local Schur complement, and can be extracted from the subdomain solves used to construct the edge approximations.

Analogously, we define the seven remaining non-zero entries by:

$$
\begin{array}{rcl@{\qquad}rcl}
(\tilde{S}_{13})_{11} & \equiv & (R_{E_1} A^{(4)} E^h \mathbf{p}_4)_1, & (\tilde{S}_{24})_{11} & \equiv & (R_{E_2} A^{(2)} E^h \mathbf{p}_4)_1 \\
(\tilde{S}_{23})_{11} & \equiv & (R_{E_2} A^{(3)} E^h \mathbf{p}_4)_1, & (\tilde{S}_{31})_{11} & \equiv & (R_{E_3} A^{(4)} E^h \mathbf{p}_1)_1 \\
(\tilde{S}_{32})_{11} & \equiv & (R_{E_3} A^{(3)} E^h \mathbf{p}_1)_1, & (\tilde{S}_{41})_{11} & \equiv & (R_{E_4} A^{(1)} E^h \mathbf{p}_1)_1 \\
(\tilde{S}_{42})_{11} & \equiv & (R_{E_4} A^{(2)} E^h \mathbf{p}_1)_1. & & &
\end{array} \tag{14}
$$

Finally, we symmetrize $\tilde{S}_{V_k}$ by the *minimum-modulus* procedure to obtain $\tilde{S}^P_{V_k}$.

THEOREM 4.3. *The vertex matrix approximations* $\tilde{S}^P_{V_k}$ *are non-singular, diagonally dominant* M*-matrices.*

Proof. See [8] for details. □

5. Numerical Results. We now present results of numerical tests on the rate of convergence of the Fourier and Probe variants of the BPS and VS algorithms. The tests were conducted for the model elliptic problem (1) for various subdomain sizes H, and fine grid sizes h. The following three coefficients were tested:

1. $a(x,y) = I$, the Laplacian, see table 1.
2. $a(x,y) = e^{10xy} I$, highly varying smooth coefficients, see table 2.
3. Highly discontinuos coefficients of Fig. 5, see table 3.

The elliptic problem was discretized using the standard five-point difference stencil, see [19], on an $(n+1) \times (n+1)$ uniform fine grid with mesh size $h = 1/n$. The subdomains were chosen to be the sub-rectangles of an $(n_s + 1) \times (n_s + 1)$ uniform coarse grid with mesh size $H = 1/n_s$. The coarse grid matrix A_H was chosen to be the five-point difference approximation of the elliptic problem on the coarse grid.

The entries of the exact solution were chosen randomly from the uniform distribution on $[-1,1]$ and the initial guess in the conjugate gradient method was chosen to be zero. The estimated condition number, κ, of the preconditioned system, and

TABLE 1
Laplace's equation: $a(x,y) = I$

h^{-1}	Ovlp	FBPS		PBPS		EVS		FVS		PVS	
H^{-1}	h/H	κ	ITN	κ	ITN	κ	ITN	κ	ITN	κ	ITN
32_2	1/16	14.3	11	9.9	9	3.4	7	5.7	11	3.2	8
32_4	1/ 8	10.0	14	7.4	11	2.6	8	4.5	11	2.5	8
32_8	1/ 4	6.4	12	5.4	11	2.5	8	3.5	10	2.4	8
64_2	1/32	19.3	12	17.1	11	4.3	7	7.2	11	4.0	9
64_4	1/16	14.5	14	11.3	12	3.4	9	5.9	13	3.2	9
64_8	1/ 8	10.3	14	8.0	12	2.8	9	4.6	12	2.7	9
64_16	1/4	6.5	13	5.6	11	2.6	8	3.6	10	2.5	8
128_2	1/64	25.0	13	31.2	13	5.5	8	9.0	11	6.5	11
128_4	1/32	19.8	16	18.4	15	4.4	10	7.4	13	4.1	10
128_8	1/16	14.7	16	12.1	13	3.5	9	5.9	13	3.4	9
128_16	1/ 8	10.4	14	8.3	13	2.8	9	4.6	11	2.7	9
128_32	1/ 4	6.5	13	5.6	11	2.6	8	3.6	10	2.5	8
256_2	1/128	31.5	13	55.9	17	6.8	9	11.0	13	11.6	13
256_4	1/64	25.4	16	33.0	19	5.5	10	9.1	13	7.2	13
256_8	1/32	19.7	16	18.5	15	4.5	10	7.3	13	4.3	10
256_16	1/16	14.7	16	12.4	13	3.5	9	5.9	13	3.3	9
256_32	1/ 8	10.4	14	8.4	13	2.8	9	4.6	11	2.7	9
256_64	1/ 4	6.5	13	5.7	11	2.6	8	3.6	10	2.4	8

the number of iterations, ITN, required to reduce the initial residual by a factor of 10^{-5} (i.e., $\|r_k\|_2/\|r_0\|_2 \leq 10^{-5}$) are listed in the tables. Unless otherwise stated, the number of nodes of overlap, N_{vs}, in the vertex regions is 1, i.e., there is one node on each vertex segment $V_k \cap E_{ij}$. The overlap ratio $\beta = h/H$ is listed as *Ovlp*.

Discussion. Tables 1 through 3 compares the performance of the various methods for the three sets of coefficients listed above. Table 1 corresponds to the Laplacian. In agreement with the theory, these results indicate that the Fourier variant FVS, has an observed rate of convergence independent of the mesh parameters H, h for fixed overlap ratio *Ovlp*. Moreover, the actual iteration numbers are quite insensitive to the choice of parameters H, h and *Ovlp*. For the range of subdomain and fine grid sizes tested, the performance of PVS is very similar to EVS. However, as the number of nodes per edge increases, it is expected that the PVS version would deteriorate, based on properties of the probe preconditioner for two subdomains in [7]. The condition numbers for the variants of the BPS algorithms grow mildly with H/h, in agreement with theory. In most cases, due to clustering of eigenvalues of the preconditioned system, the number of iterations, ITN, was often better than that predicted by the condition numbers.

Table 2 corresponds to highly varying coefficients. Here again, the results are similar to those for the Laplacian, and are in agreement with the theory. Moreover, the rate of convergence of most variants are quite insensitive to the relatively large variations in the coefficients $a(x,y)$. In order to see the importance of scaling by the coefficients, in table 2 we also tested a variant nsFVS of the FVS preconditioner, in which the edge approximations were not diagonally scaled, but were instead scaled by

TABLE 2
Highly varying coefficients: $a(x,y) = e^{10xy}I$

h^{-1}	Ovlp	FBPS		PBPS		nsFVS		FVS		PVS	
$\_H^{-1}$	h/H	κ	ITN	κ	ITN	κ	ITN	κ	ITN	κ	ITN
32_2	1/16	22.5	11	18.4	9	16.1	18	7.5	11	4.4	9
32_4	1/ 8	13.4	15	11.0	13	7.2	13	5.1	11	3.2	9
32_8	1/ 4	7.0	12	6.2	11	4.0	10	3.9	10	2.5	8
64_2	1/32	28.9	12	25.9	11	24.5	23	9.5	11	5.8	9
64_4	1/16	17.6	16	15.5	15	11.3	16	6.5	12	4.0	9
64_8	1/ 8	11.0	12	9.1	12	5.6	12	4.9	11	2.8	8
64_16	1/ 4	6.6	12	5.8	11	3.7	10	3.7	10	2.5	8
128_2	1/64	36.3	13	45.0	14	35.8	28	11.8	12	8.6	11
128_4	1/32	24.4	16	23.3	15	16.1	19	8.4	13	5.1	10
128_8	1/16	15.7	14	13.2	13	7.7	14	6.0	12	3.6	10
128_16	1/ 8	10.4	14	8.4	11	4.7	12	4.6	11	2.8	9
128_32	1/ 4	6.5	12	5.7	11	3.6	10	3.6	10	2.4	8
256_2	1/128	44.2	14	77.2	17	32.0	24	14.4	13	15.1	14
256_4	1/64	29.3	17	41.4	22	16.2	19	10.1	13	8.5	13
256_8	1/32	20.8	16	20.2	15	8.0	14	7.7	13	4.4	10
256_16	1/16	15.0	15	12.4	13	5.0	11	6.1	13	3.3	9
256_32	1/ 8	10.3	14	8.2	12	3.8	10	4.7	12	2.7	8
256_64	1/ 4	6.5	12	5.6	11	2.9	9	3.6	10	2.4	8

TABLE 3
Discontinuous coefficients: See $a(x,y)$ *of Fig. 5.*

h^{-1}	Ovlp	FBPS		PBPS		FVS		PVS	
$\_H^{-1}$	h/H	κ	ITN	κ	ITN	κ	ITN	κ	ITN
32_4	1/ 8	10.2	13	7.5	11	6.1	12	8.1	11
32_8	1/ 4	6.6	12	5.2	10	8.5	13	3.7	9
64_4	1/16	14.7	15	11.1	11	9.3	14	10.1	11
64_8	1/ 8	10.1	14	8.1	12	8.4	14	5.2	10
64_16	1/ 4	6.5	13	5.6	11	6.9	12	4.1	9
128_4	1/32	19.6	17	18.1	16	12.3	14	6.8	11
128_8	1/16	14.4	16	12.1	14	11.5	15	5.9	11
128_16	1/ 8	10.2	14	8.3	13	6.4	13	3.4	9
128_32	1/ 4	6.6	13	5.7	11	6.8	12	4.1	9
256_4	1/64	25.4	19	33.0	17	14.9	15	7.8	13
256_8	1/32	19.3	17	18.7	16	8.8	15	4.9	11
256_16	1/16	14.8	16	12.3	13	12.4	16	6.9	11
256_32	1/ 8	10.3	14	8.4	13	8.6	14	6.0	10
256_64	1/ 4	6.5	13	5.7	11	6.0	12	4.1	9

a scalar α_{ij} on each edge E_{ij}, i.e. $\tilde{S}^F_{E_{ij}} \equiv \alpha_{ij} W diag(\mu_k) W$, where $\alpha_{ij} \equiv \frac{a(x_i,y_i)+a(x_j,y_j)}{2}$, for some point $(x_i, y_i) \in \Omega_i$ and $(x_j, y_j) \in \Omega_j$. As the results indicate, this variant was much more sensitive to the variations in the coefficients.

Table 3 refers to the case of the highly discontinuous coefficients of Fig. 5. The performance is similar to the case of smooth coefficients, and the results indicate that the rate of convergence of all variants is quite insensitive to the jumps in the coefficients.

Conclusions: Both the Fourier and Probe variants of the vertex space algorithm are designed to be efficient alternatives to the original VS algorithm. Our experiments for a wide range of coefficients and grid sizes show that the efficiency does not come at a price of deteriorated performance. We hope that these variants will provide flexible and efficient methods for solving second order elliptic problems using the domain decomposition approach.

REFERENCES

[1] P. E. BJØRSTAD AND O. B. WIDLUND, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM J. Numer. Anal., 23 (1986), pp. 1093 – 1120.

[2] J. H. BRAMBLE, J. E. PASCIAK, AND A. H. SCHATZ, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp., 47 (1986), pp. 103– 134.

[3] ———, *An iterative method for elliptic problems on regions partitioned into substructures*, Math. Comp., 46 (1986), pp. 361–369.

[4] T. F. CHAN, *Analysis of preconditioners for domain decomposition*, SIAM J. Numer. Anal., 24 (1987), pp. 382–390.

[5] T. F. CHAN AND T. Y. HOU, *Domain decomposition interface preconditioners for general 2nd order elliptic problems*, Tech. Rep. CAM 88-16, Department of Mathematics, UCLA, 1990. To appear in SIAM Journal of Scientific and Statistical Computing, 1992.

[6] T. F. CHAN AND T. P. MATHEW, *An application of the probing technique to the vertex space method in domain decomposition*, in Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Periaux, and O. Widlund, eds., SIAM, 1990.

[7] ———, *The interface probing technique in domain decomposition*, Siam J. Matrix Analysis and Applications, 13 (1992).

[8] T. F. CHAN, T. P. MATHEW, AND J.-P. SHAO, *Efficient variants of the vertex space domain decomposition algorithm*, tech. rep., University of California, Los Angeles, 1992. Submitted to Siam J. Scientific and Statistical Computing.

[9] T. F. CHAN AND D. C. RESASCO, *A survey of preconditioners for domain decomposition*, Tech. Rep. /DCS/RR-414, Yale University, 1985.

[10] P. G. CIARLET, *The Finite Element Method for Elliptic Problems*, North-Holland, 1978.

[11] S. C. EISENSTAT, Personal Communication, 1985.

[12] D. FUNARO, A. QUARTERONI, AND P. ZANOLLI, *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM J. Numer. Anal., 25 (1988), pp. 1213 –1236.

[13] G. GOLUB AND D. MAYERS, *The use of preconditioning over irregular regions*, in Computing Methods in Applied Sciences and Engineering, VI, R. Glowinski and J. L. Lions, eds., Amsterdam, New York, Oxford, 1984, North-Holland, pp. 3–14. Proceedings of a conference held in Versailles, France, December 12-16,1983.

[14] G. H. GOLUB AND C. F. V. LOAN, *Matrix Computations*, Johns Hopkins Univ. Press, 1989. Second Edition.

[15] D. E. KEYES AND W. D. GROPP, *A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s166 – s202.

[16] ———, *Domain decomposition techniques for the parallel solution of nonsymmetric systems of elliptic bvps*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, 1989, SIAM.

[17] S. V. NEPOMNYASCHIKH, *On the application of the method of bordering for elliptic mixed boundary value problems and on the difference norms of $W_2^{1/2}(S)$*, Tech. Rep. 106, Computing Center of the Siberian Branch of the USSR Academy of Sciences, Novosibirsk, 1984. In

Russian.

[18] B. F. SMITH, *An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems*, Tech. Rep. 482, Department of Computer Science, Courant Institute, 1989. To Appear in Proceedings of Copper Mountain Conference on Iterative Methods, SIAM Journal of Scientific and Statistical Computing.

[19] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, 1962.

[20] O. B. 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. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, 1988, SIAM.