CHAPTER 10

An Application of the Probing Technique to the Vertex Space Method in Domain Decomposition

Tony F. C. Chan*
Tarek P. Mathew*

Abstract. We present a probing technique for constructing a modified preconditioner based on the vertex space preconditioner of Smith in R^2. The vertex space method is an optimal domain decomposition method based on many non-overlapping subdomains to solve self-adjoint 2nd order elliptic problems. This application of the probing technique significantly reduces the cost of the vertex space method and numerical tests indicate that the modified preconditioner retains the optimal convergence properties of the vertex space method.

1. Introduction. In this paper, we present a modified version of a domain decomposition method known as the vertex space (VS) method, due to Smith [7]. The VS method is an extension of the Bramble, Pasciak and Schatz (BPS) algorithm [2], and of iterative substructuring algorithms described in Widlund [8], and it is known to have a rate of convergence which is independent of the mesh size. The VS method corresponds to a block Jacobi type preconditioner (additive Schwarz [5]) for a reduced problem (Schur complement system) on the interface which separates the subdomains. The blocks correspond to nodes on the edges, the coarse grid, and certain cross-shaped subregions centered at the vertices (vertex subregions), which together constitute the interface. In order to apply the VS preconditioner, the corresponding submatrices of the Schur complement need to be computed exactly. The cost of computing the submatrices corresponding to the vertex subregions alone, is proportional to the cost of solving as many local problems on each subdomain as there are nodes in the vertex subregions lying on the boundary of the subdomain. This can, in many instances, form a significant part of the total cost of computing the solution.

Our primary motivation in modifying the VS method is to reduce this overhead cost without altering its optimal convergence properties. We obtain a modified preconditioner by replacing the submatrices by approximations. The approximations we

* 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 DAAL03-88-K-0085, by the National Science Foundation under grant FDP NSF ASC 9003002 and by the Air Force Office for Scientific Research under grant AFOSR-90-0271 (subcontract to UCLA UKRF-4-24384-90-87).

use on the coarse grid and edges are standard and have been used in many domain decomposition algorithms [2], [8]. They require no subdomain solves. The approximations we use on the vertex subregions are constructed by using the boundary probing technique of Chan and Resasco [3] and Keyes and Gropp [6]. In the case of quadrilateral subdomains, these approximations can be computed at a cost of solving about 5 local problems on each subdomain, independent of the mesh size and other VS parameters. Numerical tests of the modified preconditioner on a model problem indicate that it retains the optimal convergence of the VS preconditioner.

In the next section, we describe the elliptic problem and the Schur complement system. In Section 3, we describe the VS method in R^2. Then in Section 4, we describe the probing technique and its application to modifying the VS preconditioner. In Section 5, we present numerical results comparing the rates of convergence of various preconditioners for a model problem.

2. The elliptic problem. We consider a scalar, second order, self-adjoint elliptic equation with Dirichlet boundary conditions in a polygonal domain Ω in R^2:

$$Lu = -\nabla \cdot (a(x)\nabla u) = f(x) \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega, \tag{1}$$

where $a(x)$ is a uniformly positive definite symmetric matrix function.

The domain Ω is triangulated by a coarse mesh τ^H which consists of triangular or quadrilateral nonoverlapping elements $\Omega_1, \ldots, \Omega_N$. We refer to the coarse elements as subdomains and assume that they have diameter $O(H)$. The coarse grid nodes will be denoted by $\{x_k^H\}$. We let τ^h denote a fine grid having elements of diameter $O(h)$ which are obtained by refinement of the subdomains. The fine grid nodes will be denoted $\{x_k^h\}$.

To obtain a discretisation of L, we use either finite elements or finite differences on τ^h. A_h will denote the stiffness matrix on the fine grid τ^h and A_H will denote the restriction of A_h to τ^H. The interface Γ is defined to be the union of the subdomain boundaries in the interior of the domain Ω: $\Gamma \equiv \cup_j(\partial\Omega_j) \cap \Omega$.

Ordering the unknowns in the interior of Ω_1 in u_1, followed by those in the interior of Ω_2 in u_2, and so on till we group the unknowns on Γ in u_Γ, we obtain the following $(N+1) \times (N+1)$ block structure for A_h in the discretisation of (1):

$$\begin{pmatrix} A_{11} & & 0 & A_{1\Gamma} \\ & \ddots & & \vdots \\ 0 & & A_{NN} & A_{N\Gamma} \\ A_{1\Gamma}^T & \cdots & A_{N\Gamma}^T & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_N \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_N \\ f_\Gamma \end{pmatrix}. \tag{2}$$

Eliminating the unknowns in the interior of the subdomains, i.e., $u_1, \ldots, u_N$, we obtain the reduced system for u_Γ:

$$Su_\Gamma = f_\Gamma - \sum_{i=1}^{N} A_{i\Gamma}^T A_{ii}^{-1} f_i, \tag{3}$$

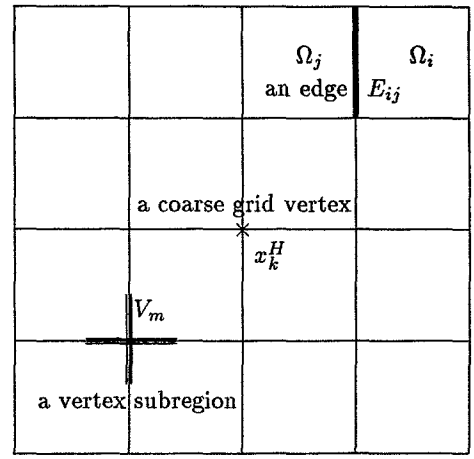
where the Schur complement S is defined by $S \equiv A_{\Gamma\Gamma} - \sum_{i=1}^{N} A_{i\Gamma}^T A_{ii}^{-1} A_{i\Gamma}$. In this paper we focus on solving problem (3) on the interface Γ by using the preconditioned conjugate gradient method, with preconditioner M. We note that we may construct a preconditioner for the entire matrix A_h based on a preconditioner for the Schur

complement S, see [2]. Once u_Γ is determined, the complete solution in the interior of the subdomains is obtained from:

$$u_i = A_{ii}^{-1}(f_i - A_{i\Gamma}u_\Gamma), \quad \text{for } i = 1, \ldots, N.$$

Remark. The Schur complement S is seldom computed since it is quite expensive to compute, involving the inversion of A_{ii} on each subdomain n_i times, where n_i is the number of nodes on the boundary of Ω_i. One of the well known advantages of using the preconditioned conjugate gradient method is that with a choice of efficient

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



preconditioner it is possible to compute the solution to (3) at a cost proportional to a small number $m \ll n_i$ of matrix vector products involving S (each product of S involves the inversion once of each A_{ii}).

3. The Vertex Space Method. The vertex space method (VS) of Smith [7] provides a preconditioner M_{VS} for the Schur complement system (3). It is based on the BPS algorithm [2], and iterative substructuring algorithms [8]. However, the VS preconditioner is known to have $\kappa(M_{VS}^{-1}S) = O(1)$ independent of H and h, which is an improvement over the BPS algorithm which has $\kappa(M_{BPS}^{-1}S) = O(1 + \log^2(H/h))$.

Each iteration of the preconditioned conjugate gradient method to solve (3) involves a matrix vector product with S (and each product of S involves one solve on each subdomain) as well as a matrix vector product with M_{VS}^{-1}. The VS preconditioner M_{VS} is never formed explicitly. Rather, given a grid function g_Γ, the action of M_{VS}^{-1} on this grid function is defined as a sum of three components, each of which depends on g_Γ. The basic idea in constructing each component of $M_{VS}^{-1}g_\Gamma$ is to restrict the forcing term g_Γ to certain subregions of the interface Γ and to solve local problems on these subregions with submatrices of the Schur complement S as coefficient matrices. The choice of subregions is important, and the VS method provides a particular choice of subregions which leads to an optimal algorithm.

The subregions of Γ used in the VS method in R^2 are the following, see Figure 1:

1. The *coarse grid* consisting of the vertices $\{x_k^H\}$.
2. The *edges* $\{E_{ij}\}$, where E_{ij} forms the common boundary to Ω_i and Ω_j without including the endpoints.
3. The *vertex subregions*, $\{V_k\}$, where each V_k is a star-shaped piece centered at coarse grid vertex x_k^H and consists of the vertex x_k^H and segments of length $O(H)$ of all edges E_{ij} emanating from x_k^H. In the case of quadrilateral subdomains, each vertex segment V_k is cross-shaped and can be partitioned into the following 5 disjoint pieces: one coarse grid node x_k^H and four *edge segments* (two horizontal and two vertical segments of edges E_{ij} emanating from x_k^H; we assume that none of the *edge segments* contain the coarse grid vertex x_k^H).

The following restriction and interpolation (extension) maps are defined for grid functions on Γ and its subregions.

1. *Coarse grid interpolation and restriction map.* Based on the coarse grid, we let R_H^T denote the standard interpolation map which maps coarse grid functions on Γ into fine grid nodes on Γ. For instance, if piecewise linear finite element functions are used in the coarse mesh τ^H, then R_H^T consists of linear interpolation on the edges E_{ij} of the coarse grid values on $\{x_k^H\}$. We let R_H denote the transpose of R_H^T and it corresponds to the restriction of fine grid functions defined on Γ to the coarse grid nodes on Γ.

2. *Edge restriction and extension maps.* Given a grid function g_Γ on Γ, we let $R_{E_{ij}} g_\Gamma$ denote the restriction of the nodal values of g_Γ to the nodes on the interior of the edge E_{ij}. Thus:

$$(R_{E_{ij}} g_\Gamma)(x_k^h) \equiv \begin{cases} g_\Gamma(x_k^h) & \text{if } x_k^h \text{ is an interior node in } E_{ij} \\ 0 & \text{if } x_k^h \text{ is not an interior node in } E_{ij} \end{cases}$$

Its transpose $R_{E_{ij}}^T$ denotes the extension of nodal values in the interior of E_{ij} to the rest of Γ such that it is zero outside E_{ij}.

3. *Vertex subregion restriction and extension maps.* Given a grid function g_Γ, we let $R_{V_k} g_\Gamma$ denote the restriction of nodal values of g_Γ to the nodes in the vertex segment V_k:

$$(R_{V_k} g_\Gamma)(x_k^h) \equiv \begin{cases} g_\Gamma(x_k^h) & \text{if } x_k^h \text{ is a node in } V_k \\ 0 & \text{if } x_k^h \text{ is not a node in } V_k \end{cases}$$

Its transpose $R_{V_k}^T$ denotes the extension of nodal values in V_k to the rest of Γ, such that it is zero outside V_k.

For the three subregions, we define the following coefficient matrices:

$$A_H \equiv R_H A_h R_H^T, \quad S_{E_{ij}} \equiv R_{E_{ij}} S R_{E_{ij}}^T, \quad S_{V_k} \equiv R_{V_k} S R_{V_k}^T,$$

where A_H is a coarse grid operator, and $S_{E_{ij}}$ and S_{V_k} are submatrices of S obtained by picking off elements of S corresponding to the nodes in E_{ij} and V_k respectively. The action of the inverse of the VS preconditioner M_{VS} can now be defined:

$$M_{VS}^{-1} g_\Gamma \equiv \left(R_H^T A_H^{-1} R_H + \sum_{ij} R_{E_{ij}}^T S_{E_{ij}}^{-1} R_{E_{ij}} + \sum_k R_{V_k}^T S_{V_k}^{-1} R_{V_k} \right) g_\Gamma. \tag{4}$$

Of course, the inverses of the submatrices A_H, $S_{E_{ij}}$ and S_{V_k} need never be formed, only their actions need be computed. This is often done by using direct solvers.

Remark. The matrix A_H can be shown to equal $S_H \equiv R_H S R_H^T$, for some model problems, and in those cases, the VS preconditioner exactly corresponds to the additive Schwarz preconditioner [5] based on the subregions of Γ.

The VS preconditioner is known to have an optimal rate of convergence under certain conditions:

THEOREM 3.1. *If the overlap between V_k and the edges is $O(H)$, then $\kappa(M_{VS}^{-1}S) = O(1)$, independent of H and h.*

Proof. See [7]. □

4. Probed Vertex Space method. In this section, we replace the VS preconditioner M_{VS} by an approximation M_{PVS} to solve equation (3). M_{PVS} is obtained by replacing the submatrices A_H, $S_{E_{ij}}$ and S_{V_k} used in the VS preconditioner (4) by approximations $\tilde{A}_H$, $\tilde{S}_{E_{ij}}$ and $\tilde{S}_{V_k}$ respectively:

$$M_{PVS}^{-1} g_\Gamma \equiv \left(R_H^T \tilde{A}_H^{-1} R_H + \sum_{ij} R_{E_{ij}}^T \tilde{S}_{E_{ij}}^{-1} R_{E_{ij}} + \sum_k R_{V_k}^T \tilde{S}_{V_k}^{-1} R_{V_k} \right) g_\Gamma. \tag{5}$$

The resulting probed preconditioner M_{PVS} will cost about the same to invert per iteration as the vertex space preconditioner M_{VS}. However, the overhead cost of setting up M_{PVS} will be less than for M_{VS}, i.e., the cost of computing $\tilde{A}_H$, $\tilde{S}_{E_{ij}}$ and $\tilde{S}_{V_k}$ will be significantly less than for computing A_H, $S_{E_{ij}}$ and S_{V_k}.

A desirable property we look for in the approximations $\tilde{A}_H$, $\tilde{S}_{E_{ij}}$ and $\tilde{S}_{V_k}$ is that they be spectrally equivalent to A_H, $S_{E_{ij}}$ and S_{V_k} respectively, with respect to variations in mesh size, since then it can be easily shown that the modified preconditioner M_{PVS} would be spectrally equivalent to M_{VS}. In this section, we will primarily focus on finding approximations to the *vertex submatrices* S_{V_k}, since there are many well known ways of approximating A_H and $S_{E_{ij}}$ [2], [8]. However, for completeness, we briefly describe the approximations $\tilde{A}_H$ and $\tilde{S}_{E_{ij}}$ we used in our numerical tests.

4.1. Approximation of the coarse grid operator A_H. A simple choice would be to replace the coarse grid operator by the coarse grid Laplacian, $\tilde{A}_H = -\Delta_H$. For variable coefficient problems, we define $\tilde{A}_H$ to be the coarse grid discretisation of L, if finite difference methods are used. In case a finite element method is used, we let $\tilde{A}_H$ be the coarse grid stiffness matrix computed using numerical integration on the coarse mesh elements.

4.2. Approximation of the edge matrices $S_{E_{ij}}$. The entries of $S_{E_{ij}}$ represent the coupling in S between nodes on the edge E_{ij}. Approximation of the edge matrices $S_{E_{ij}}$ have received much attention, especially for the case of two subdomains [4], [3]. In our tests, as in the BPS algorithm [2], we define each $n \times n$ matrix $\tilde{S}_{E_{ij}}$ to be:

$$\tilde{S}_{E_{ij}} \equiv \alpha_{ij} W D W^{-1}, \qquad \text{where } W_{ij} = \sqrt{\frac{2}{n+1}} \sin\left(\frac{ij\pi}{n+1}\right),$$

and where D is a diagonal matrix with entries $D_{ii} = 2\sin(i\pi/2(n+1))$. The positive scalar coefficient

$$\alpha_{ij} \equiv \frac{\alpha_i + \alpha_j}{2}, \text{ where } \alpha_i \equiv \frac{\lambda_{min}(a(x_i)) + \lambda_{max}(a(x_i))}{2},$$

for some point $x_i \in \Omega_i$. Here $\lambda(a(x))$ represents the eigenvalues of the 2×2 coefficient

matrix $a(x)$ at x. Note that W is a discrete sine transform, and so $\tilde{S}_{E_{ij}}$ can be inverted in $O(n\log(n))$ operations using FFT's. It can be shown that the resulting matrix $\tilde{S}_{E_{ij}}$ is spectrally equivalent to $S_{E_{ij}}$ for large n, see [2], [1].

4.3. Approximation of the vertex submatrices S_{V_k}. The entries of S_{V_k} represent the coupling in S between the nodes in V_k. Our approach in computing approximations to these vertex submatrices is based on the principles used in the probing technique of Chan and Resasco [3] and Keyes and Gropp [6]. We briefly sketch the basic idea, and then illustrate the procedure in matrix terms.

It is known that the entries of S decay rapidly in magnitude with increasing distance between nodes. When constructing $\tilde{S}_{V_k} \approx S_{V_k}$, we would like to define the entries of $\tilde{S}_{V_k}$ to be zero corresponding to "small" entries in S_{V_k}. However, since S_{V_k} is not explicitly known, we will assume a sparsity pattern for $\tilde{S}_{V_k}$. Then, to determine the nonzero entries of $\tilde{S}_{V_k}$, which are to approximate the corresponding entries in S_{V_k}, we multiply a few carefully chosen probe vectors $p_1, \ldots, p_m$ (defined locally on V_k with $m \ll \dim(S_{V_k})$) by both $\tilde{S}_{V_k}$ and S_{V_k}, and equate the products. In other words, we look for an approximation $\tilde{S}_{V_k}$ with a given sparsity pattern such that

$$\tilde{S}_{V_k} p_i = S_{V_k} p_i, \quad \text{for } i = 1, \ldots, m. \tag{6}$$

In this paper we choose $m = 5$. We will discuss the construction of $\tilde{S}_{V_k}$ later in algorithm 2 of Section 4.3.3. But at this point, we would like to summarise the overall procedure for computing all the vertex submatrices. This will be done concurrently on all the vertex subregions as follows:

Algorithm 1. (To compute all $\tilde{S}_{V_k}$)

1. Define $v_1, \ldots, v_5$ on Γ by $v_i = p_i$ on each V_k, with $v_i = 0$ outside $\cup_k V_k$.
2. Form the products $Sv_1, \ldots, Sv_5$.
3. Find the restriction of Sv_i to each vertex subregion V_k, namely, $R_{V_k} S v_i$.
4. For each V_k, determine $\tilde{S}_{V_k}$ using algorithm 2 with data $q_1, \ldots, q_5$, where $q_i \equiv R_{V_k} S v_i$.

In the following subsections, we discuss the details of step 4.

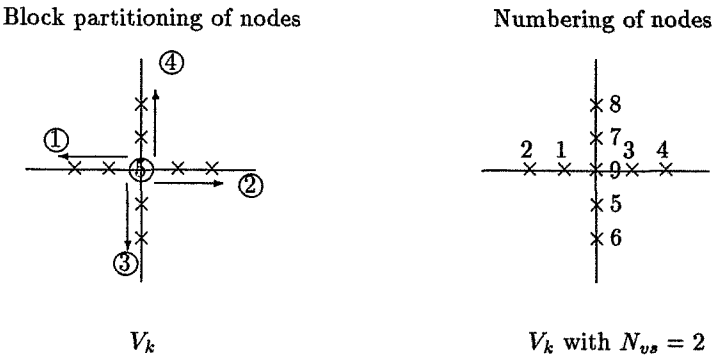
4.3.1. Assumptions on the sparsity pattern of $\tilde{S}_{V_k}$. As mentioned before, the procedure to compute $\tilde{S}_{V_k}$ will depend on its sparsity pattern and on the choice of the probe vectors. Our assumptions about the decay of elements in S_{V_K} (and hence the sparsity pattern of each $\tilde{S}_{V_k}$) are:

1. *Within each edge segment of V_k, we assume that the coupling is strong only between adjacent nodes.* This assumption leads us to choose the submatrices of $\tilde{S}_{V_k}$ corresponding to the couplings within each edge segment to have at most three nonzero entries in each row. (In the natural ordering, this corresponds to a tridiagonal matrix.)
2. *We assume that the couplings between the different edge segments is not strong except between the nodes which are closest (adjacent) to the vertex node x_k^H.* This assumption leads us to choose the submatrices of $\tilde{S}_{V_k}$ corresponding to the coupling between different edge segments to have zero entries, except for the entries corresponding to the coupling between the nodes closest to the vertex.

Given an ordering of the nodes on each vertex subregion V_k, the precise sparsity pattern of $\tilde{S}_{V_k}$ can be easily determined from the above assumptions.

In order to illustrate this procedure, we will show the construction explicitly in matrix terms for cross-shaped vertex subregions V_k. We assume the following ordering of nodes within each V_k having N_{vs} nodes on each of its edge segments (with a total of $(4N_{vs}+1)$ nodes on V_k):

FIG. 2. *Ordering of unknowns within each vertex subregion V_k*



1. The nodes on the left horizontal edge segment are numbered $1, \ldots, N_{vs}$, with the numbers increasing in the direction away from the vertex x_k^H.
2. Similarly, the nodes on the right horizontal edge segment are numbered from $N_{vs}+1, \ldots, 2N_{vs}$, increasing in the direction away from x_k^H.
3. The nodes on the bottom vertical edge segment are numbered $2N_{vs}+1, \ldots, 3N_{vs}$.
4. The nodes on the top vertical edge segment are numbered $3N_{vs}+1, \ldots, 4N_{vs}$.
5. Finally, the vertex node x_k^H is numbered $4N_{vs}+1$.

This ordering is illustrated in Figure 2 for the case of a vertex segment V_k containing 2 nodes on each edge segment (i.e., $N_{vs} = 2$), and a total of 9 nodes. The arrow indicates the direction in which the node numbers increase.

Corresponding to each of the 5 pieces constituting V_k, we partition the nodes on V_k into 5 blocks. This results in a 5×5 block structure for each $\tilde{S}_{V_k}$:

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

where each $\tilde{S}_{ij}$ is a submatrix corresponding to the coupling in S of nodes in block i and block j. Note that the submatrices $\tilde{S}_{12}$ and $\tilde{S}_{34}$ and their transposes are zero, since it can be easily shown that there is no coupling between edges 1 and 2 and between edges 3 and 4 in S_{V_k}. Moreover, based on assumption 1 on the sparsity of entries, the first four diagonal blocks $\tilde{S}_{11}, \ldots, \tilde{S}_{44}$ are chosen to be tridiagonal. Similarly, based on assumption 2, we choose the submatrices $\tilde{S}_{13}$, $\tilde{S}_{14}$, $\tilde{S}_{23}$ and $\tilde{S}_{24}$ and their transposes to have zero entries everywhere except in location $(1,1)$:

$$\tilde{S}_{ij} = \begin{pmatrix} (\tilde{S}_{ij})_{11} & 0 & \cdots \\ 0 & 0 & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}, \quad \text{for } i = 1,2; \quad j = 3,4.$$

Since we have not assumed any sparsity pattern for the coupling between the vertex x_k^H and the remaining nodes on V_k, the submatrices $\tilde{S}_{i5}$ and $\tilde{S}_{i5}^T$, for $i = 1,\ldots,5$, will be assumed to be dense.

4.3.2. Choice of local probe vectors. We now describe our local probe vectors $p_1,\ldots,p_5$ on each V_k in terms of $e_{10} = (1,0,1,0,\ldots)^T$ and $e_{01} = (0,1,0,1,\ldots)^T$, which are each vectors of length N_{vs} (the number of nodes per edge segment). Then, for the block partitioning of nodes on V_k, we define:

$$(7) \quad p_1 \equiv \begin{pmatrix} e_{10} \\ e_{01} \\ 0 \\ 0 \\ 0 \end{pmatrix}, p_2 \equiv \begin{pmatrix} e_{01} \\ e_{10} \\ 0 \\ 0 \\ 0 \end{pmatrix}, p_3 \equiv \begin{pmatrix} 0 \\ 0 \\ e_{10} \\ e_{01} \\ 0 \end{pmatrix}, p_4 \equiv \begin{pmatrix} 0 \\ 0 \\ e_{01} \\ e_{10} \\ 0 \end{pmatrix}, p_5 \equiv \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

4.3.3. Construction of $\tilde{S}_{V_k}$. Given five vectors $q_1,\ldots,q_5$ of length $4N_{vs}+1$, we wish to construct a matrix $\tilde{S}_{V_k}$ of size $4N_{vs}+1$ having the the assumed sparsity pattern and satisfying $\tilde{S}_{V_k}p_i = q_i$ for $i = 1,\ldots,5$. In matrix form, $\tilde{S}_{V_k}[p_1,\ldots,p_5] = [q_1,\ldots,q_5]$ yeilds:

$$(8) \quad \begin{pmatrix} \tilde{S}_{11}e_{10} & \tilde{S}_{11}e_{01} & \tilde{S}_{13}e_{10} & \tilde{S}_{14}e_{10} & \tilde{S}_{15} \\ \tilde{S}_{22}e_{01} & \tilde{S}_{22}e_{10} & \tilde{S}_{23}e_{10} & \tilde{S}_{24}e_{10} & \tilde{S}_{25} \\ \tilde{S}_{13}^T e_{10} & \tilde{S}_{23}^T e_{10} & \tilde{S}_{33}e_{10} & \tilde{S}_{33}e_{01} & \tilde{S}_{35} \\ \tilde{S}_{14}^T e_{10} & \tilde{S}_{24}^T e_{10} & \tilde{S}_{44}e_{01} & \tilde{S}_{44}e_{10} & \tilde{S}_{45} \\ \tilde{S}_{15}^T e_{10} & \tilde{S}_{25}^T e_{10} & \tilde{S}_{35}^T e_{10} & \tilde{S}_{45}^T e_{10} & \tilde{S}_{55} \end{pmatrix} = [q_1,\ldots,q_5],$$

where the entries of $\tilde{S}_{V_k}p_i$ have been simplified using the fact that $\tilde{S}_{ij}e_{01}$ are zero for $i = 1,2$ and $j = 3,4$.

Corresponding to each of the $5(4N_{vs}+1)$ entries in (8), there is one equation for the $12N_{vs}+1$ nonzero unknowns in the upper triangle of $\tilde{S}_{V_k}$. This results in an overdetermined system. If each column q_i in (8) satisfies $q_i = S_{V_k}p_i$ for a matrix S_{V_k} having the same sparsity pattern as $\tilde{S}_{V_k}$, then these equations are easily shown to be consistent, and S_{V_k} can easily be reconstructed. For general right hand side $[q_1,\ldots,q_5]$, equation (8) does not lead to a consistent system and so we specify a procedure for computing an approximate solution to (8), which equals the unique solution when it exists and which and also guarantees the symmetry of $\tilde{S}_{V_k}$.

This construction of $\tilde{S}_{V_k}$ is based on three observations. One is that for tridiagonal matrices M (in our application $M = \tilde{S}_{ii}$, for $i = 1,\ldots,4$), all its nonzero entries, and hence M can be determined from Me_{10} and Me_{01}. The algorithm we use to determine such tridiagonal M was first proposed in [6], and we refer to it as the Symmetric-probe algorithm. It is based on the following illustration:

$$(9) \quad \begin{pmatrix} M_{11} & M_{12} & & & \\ M_{12} & M_{22} & M_{23} & & \\ & M_{23} & M_{33} & M_{34} & \\ & & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \vdots & \vdots \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{12}+M_{23} & M_{22} \\ M_{33} & M_{23}+M_{34} \\ \vdots & \vdots \end{pmatrix},$$

from which it is clear that the elements M_{ij} can be recovered from the matrix vector products. The second observation is that since for $i = 1, 2$ and $j = 3, 4$ the submatrices $\tilde{S}_{ij}$ contain only one element in the $(1,1)$ entry, this is easily found to be $(\tilde{S}_{ij})_{11} = (\tilde{S}_{ij}e_{10})_1$. The third observation is that the entries in the last column of $\tilde{S}_{V_k}$, namely $\tilde{S}_{i5}$ for $i = 1, \ldots, 5$, are exactly equal to $\tilde{S}_{V_k}p_5$. This completes the upper triangle of $\tilde{S}_{V_k}$. The rest of the submatrices are obtained by symmetry.

Note that all the quantities we have used, such as $\tilde{S}_{ii}e_{10}$, $\tilde{S}_{ii}e_{01}$ for $i = 1, \ldots, 4$, $\tilde{S}_{ij}e_{10}$ for $i = 1, 2$ and $j = 3, 4$ and $\tilde{S}_{i5}$ for $i = 1, \ldots, 5$ are present in equation (8). We summarise the algorithm in the following:

Algorithm 2. (To construct $\tilde{S}_{V_k}$ from $q_1, \ldots, q_5$ in step 4 of algorithm 1)

1. For $i = 1, \ldots, 4$, determine a tridiagonal $\tilde{S}_{ii}$ from $\tilde{S}_{ii}e_{10}$ and $\tilde{S}_{ii}e_{01}$ (available from $q_1, \ldots, q_4$, see equation (8)) using the Symmetric-probe algorithm.
2. For $i = 1, 2$ and $j = 3, 4$, obtain the only nonzero entry in the submatrices $\tilde{S}_{ij}$, namely $(\tilde{S}_{ij})_{11} = (\tilde{S}_{ij}e_{10})_1$ (available in q_3, q_4)
3. The last column of $\tilde{S}_{V_k}$ is equal to q_5.
4. The entries in the lower triangular part of $\tilde{S}_{V_k}$ are obtained by taking the transpose of the blocks in the upper triangular part.

Computing all the vertex submatrix approximations $\tilde{S}_{V_k}$ using algorithm 1 requires 5 solves on each subdomain. In contrast, computing the vertex submatrices S_{V_k} requires $8N_{vs} + 4$ solves on each subdomain (for quadrilateral subdomains), since there are $8N_{vs} + 4$ nodes on the boundary of each subdomain which lie on the vertex subregions. Thus the probed VS method can cost substantially less if N_{vs} is large.

Remark. For 5 point difference schemes on quadrilateral subdomains, the last row and column of S_{V_k} is exactly equal to the last row and column of the matrix A_{V_k}, $(\equiv R_{V_k}A_hR_{V_k}^T)$, i.e., the matrix A_h restricted to the nodes on V_k. This can be derived from the fact that for such discretisations, the corner nodes in the quadrilateral domains do not influence the solution. In such cases, the fifth probe p_5 is not required.

Remark. If $N_{vs} = 1$ or 2, then the Symmetric-probe algorithm is not required in the construction of each $\tilde{S}_{V_k}$. For instance, if $N_{vs} = 1$, then the blocks $\tilde{S}_{ii}$ are scalars. If $N_{vs} = 2$, then $\tilde{S}_{ii}$ are 2×2 matrices, for $i = 1, \ldots, 4$ and $\tilde{S}_{ii}e_{10}$ and $\tilde{S}_{ii}e_{01}$ form the first and second columns of $\tilde{S}_{ii}$. Experiments conducted in [7], as well by us indicate that $N_{vs} = 1$ or 2 is usually sufficient.

5. Numerical Results. We present here the results of tests done using three preconditioners to solve the Schur complement system (3). In these preliminary tests, we consider only the 5-point Laplacian on a square uniform grid with mesh width $h = 1/N$. The domain was divided into equally sized subdomains of length $H = 1/N_c$, and the vertex subregions V_k were chosen to have N_{vs} nodes on each edge segment.

The three preconditioners we consider are the BPS preconditioner [2]:

$$M_{BPS}^{-1}g_\Gamma \equiv \left(R_H^T \tilde{A}_H^{-1} R_H + \sum_{ij} R_{E_{ij}}^T \tilde{S}_{E_{ij}}^{-1} R_{E_{ij}} \right) g_\Gamma, \tag{10}$$

the VS preconditioner [7] and the PVS preconditioner, which we have described in this paper. In all three preconditioners, we used $\tilde{A}_H = \Delta_H$, the coarse grid Laplacian. In addition, we used $\tilde{S}_{E_{ij}}$ to be the Dryja preconditioner, WDW^{-1} described earlier. Note that $\alpha_{ij} = 1$ for $L = -\Delta$. For the VS preconditioner, the vertex submatrices S_{V_k} were computed exactly. For the PVS preconditioner, the matrices $\tilde{S}_{V_k}$ were computed

TABLE 1
PROBING THE VERTEX SPACE MATRICES.

FINE GRID	SUBDOMAINS	VERTEX	BPS	VS	PVS
$N \times N$	$N_c \times N_c$	SIZE N_{vs}	ITER.	ITER.	ITER.
16×16	2×2	2	10	14	11
32×32	2×2	2	12	13	12
32×32	4×4	2	20	13	12
64×64	2×2	2	13	14	12
64×64	4×4	2	22	16	13
64×64	8×8	2	21	14	13
128×128	4×4	2	24	17	14
128×128	8×8	2	23	16	13
128×128	16×16	2	21	14	13
16×16	2×2	1	10	12	11
32×32	2×2	1	12	13	11
32×32	4×4	1	20	13	12
64×64	2×2	1	13	14	13
64×64	4×4	1	22	16	14
64×64	8×8	1	21	14	12
128×128	4×4	1	24	17	16
128×128	8×8	1	23	16	14
128×128	16×16	1	21	14	12

using the probing technique we have described. In table 1 we list the number of iterations required to reduce the 2-norm of the residual by a factor of 10^{-5} for various grid sizes h, subdomain sizes H and vertex sizes N_{vs}. The initial guess was chosen to be a vector of all 1's, and the right hand side in (2) was chosen to be 0.

We note that the iteration count for both the VS method and the PVS method are nearly the same, showing that the probing approximation does not destroy the optimal convergence rates of the VS method. Both have asymptotically better convergence rates than the BPS algorithm, and these results agree with the theoretical results for the convergence rates of the BPS and VS methods. However, the VS and PVS method involve additional overhead cost, which costs about 5 additional iterations for the PVS version, and about $8N_{vs}$ more iterations for the VS method. The cost per iteration of the VS and PVS methods are the same since we have not used sparse solvers for the $\tilde{S}_{V_k}$, but both are slightly more expensive than BPS due to the inversion of the vertex matrices $\tilde{S}_{V_k}$. Thus, unless the iteration count of the BPS algorithm exceeds the iteration count of the PVS method by 5 or more, the BPS method may be more efficient. However, since asymptotically the BPS algorithm has a logarithmic growth in its condition number, there is a cross-over point after which the VS and PVS methods are more efficient.

Acknowledgements. The authors would like to express their gratitude to Dr. Charles Tong for his kind assistance with the numerical tests conducted in this paper.

REFERENCES

[1] Petter E. Bjørstad and Olof B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23(6):1093 – 1120, 1986.

[2] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.*, 47(175):103– 134, 1986.

[3] Tony F. Chan and Diana C. Resasco. A survey of preconditioners for domain decomposition. Technical Report /DCS/RR-414, Yale University, 1985.

[4] Maksymilian Dryja. A capacitance matrix method for Dirichlet problem on polygon region. *Numer. Math.*, 39:51 – 64, 1982.

[5] Maksymilian Dryja and Olof B. Widlund. Some domain decomposition algorithms for elliptic problems. Technical Report 438 also Ultracomputer Note 155, Department of Computer Science, Courant Institute, 1989. To appear in the proceeding of the Conference on Iterative Methods for Large Linear Systems held in Austin, Texas, October 1988, to celebrate the Sixty-fifth Birthday of David M. Young, Jr., Academic Press, Orlando, Florida, 1989.

[6] David E. Keyes and William D. Gropp. A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation. *SIAM J. Sci. Stat. Comput.*, 8(2):s166 – s202, 1987.

[7] Barry F. Smith. An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems. Technical Report 482, Department of Computer Science, Courant Institute, 1989. To Appear in Proceedings of Copper Mountain Conference on Iterative Methods, April 1990.

[8] Olof B. Widlund. Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, 1988. SIAM.